К ВАШИМ УСЛУГАМ

Каждый хочет быть чуточку более фотогеничным. В этом месяце мы задали членам редакции вопрос: «Как бы вы изменили свое фото в Gimp, если бы вам случилось этим заняться»

Пол Хадсон
«Никак. Я прекрасен сам по себе»

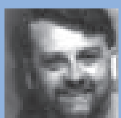
Грэм Моррисон
«Я бы взял пипетку и нарисовал слезы, как у клоуна»

Эфрейн Хернандез-Мендоза
«Я бы сделал его цветным. Пусть все видят, насколько глубоки мои латинские корни»

Майк Сондерс
«Я бы применил волшебный фильтр, который придал бы моей коже такой оттенок, как будто она видела солнечный свет каждый день в течение полгода»

Ребекка Смолли
«Распушила бы себя 12 пикселей. Не думаю, чтобы это что-то изменило на фотографии, но ощущения стали бы приятнее»

Эндрью Грегори
«На моем старом ПК, на чердаке, есть такой JPEG. Каждый раз, как я смотрю на него, там добавляется морщин, хотя я остаюсь прежним. Странно все это»



Нейл Ботвик
«Gimp недостаточно мощен. Я бы использовал rm»

Дэвид Картрайт
«Я бы добавил себе морщин, седые волосы и бороду, чтобы отличаться от тех, чьи фотографии вы видите на этой странице»

Энди Ченнел
«Растянул бы по вертикали. В реальности я гораздо стройнее»

Ричард Коббет
«Рассеянные линзы, и побольше, чтобы скрыть трещины на объективе»

Энди Хадсон
«Мне бы пригодился фильтр 'Remove Five O'Clock Shadow'. Он делает меня более презентабельным»

Ричард Смедли
«Кому нужен Gimp? Дайте мне AA-lib и я переведу свое фото в ASCII art. Вот, так намного лучше!»

В защиту mcedit

Так уж получилось, что Linux Format, успешно избежавший войны браузеров, оказался невольно вовлечен в войну текстовых редакторов. Причем – не без моего участия.

Месяц назад мы опубликовали учебник Emacs и решили, что будет неправильно, если мы не дадим высказаться его давним оппонентам – пользователям *Vi*. Но ведь в Linux есть и другие текстовые редакторы!

Сегодня я хочу злоупотребить выделенной мне колонкой и высказаться в защиту mcedit – того самого редактора, который возникает при нажатии F4 в Midnight Commander. Да, он не так стар, как *Vi*, и к нему не приложил руку Ричард Столлмен, как к Emacs (хотя Мигель де Икаса тоже чего-то стоит). Но чем он хуже? Как и свои большие братья, mcedit умеет вставлять и удалять символы, подсвечивать синтаксис, выполнять поиск по регулярному выражению, проверять правописание, отправлять почту, выделять парные скобки и «прыгать» между ними, форматировать абзацы, записывать макросы и даже активно использует нечеловеческое (для владельцев современных клавиатур) сочетание Meta+<что-то>. Работа с буфером обмена достойна порицания (особенно если речь заходит об X Window), но вдумайтесь – разве это не немой призыв оставаться в консоли, как подобает истинному хакеру? Любителей делать все, не выходя из текстового редактора (да, я про вас говорю, пользователи Emacs!) тоже можно порадовать – в mcedit встроен неплохой файловый менеджер. Для переключения нужно нажать Esc, а чтобы попасть обратно – F4.

Ну что, понравилось? Если да – добро пожаловать в клуб! Наверняка mcedit уже есть в вашем дистрибутиве и вы можете начать пользоваться им незамедлительно, не запоминая магических команд. Если нет – прекрасно, работайте в том редакторе, который считаете самым удобным. В конце концов, текстовый редактор – это всего лишь инструмент, позволяющий делать какую-то работу, вот и займитесь ею. А войны и распри оставьте другим.

С уважением,
СИНИЦЫН ВАЛЕНТИН
главный редактор LinuxFormat в России

КАК С НАМИ СВЯЗАТЬСЯ

Письма для публикации:
letters@linuxformat.ru
Подписка и предыдущие номера:
subscribe@linuxformat.ru
Техническая поддержка:
answers@linuxformat.ru
Проблемы с дисками:
disks@linuxformat.ru
Общие вопросы: **info@linuxformat.ru**
Website: **www.linuxformat.ru**

Адрес редакции: Россия, Санкт-Петербург, ул. Гончарная, 23, офис 54
Телефон редакции: (812) 717-00-37
Дополнительная информация на стр.128

МИССИЯ ЖУРНАЛА

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО

Содержание

Добро пожаловать в LinuxFormat – ваш гид в мире Linux!

LINUX FORMAT

LXF4(78) Апрель 2006

» ЧИТАЙТЕ В ЭТОМ ВЫПУСКЕ

ФОТОГРАФИРУЕМ С LINUX 46

Пособие по использованию программ с нашего диска для редактирования, публикации и обмена фотографиями

10 Тестирование SUSE 10.1
Хакеры, познакомьтесь – чайники. Вы сработаетесь...

42 Web 2.0
Наслышаны? Представляем бета-версию нового Интернета

54 Syllable: простая и понятная
История восхитительной настольной ОС

88 Утилиты командной строки
По просьбам трудящихся

20 Лучшие DVD-проигрыватели... навсегда!

МЫСЛИ ВСЛУХ »

«Я поделился с вами каждой крупицей знаний о Gimp, и теперь пора переходить к другим темам.»

80 Прощальное слово Майкла Дж. Хэммела

«Приятно бросить вызов Apple, но я не вижу причин для дальнейшего роста доли Linux.»

54 Продолжайте, Кристиан ван дер Влиет

12 Yellow Dog Linux

04 **Новости**
GCC 4.1, GNOME 2.14, ...

10 **SUSE 10.1**
GCC 4.1. KDE 3.5. Xen 3.0...
Yast-zee!

12 **Yellow Dog Linux 4.1**
Дистрибутив для PPC жив-здоров

14 **Gentoo Linux 2006.0**
Что случилось с Portage?

15 **Arkeia 5.4.9**
Резервирование начинается здесь

16 **Airline Tycoon Deluxe**
Оригинальный полетный симулятор

18 **Distrowatch**
В защиту Slackware

20 **Сравнение: DVD-проигрыватели**
Apt Get Carter, Tux Soup, Rocky Vi...

26 **Выбираем прокси-сервер**
Squid или Ooops!?

30 **HotPicks**
Только что с конвейера

36 **Что такое... Jingle?**
Linux заставит вас говорить

38 **Mono-глоты**
Борнштейн и Дамбилл – о Mono

42 **Web 2.0**
Все по-настоящему

46 **Фотографируйте с Linux**
Трюки и приемы

54 **Syllable**
Лучше Linux?

60 **Интервью LXF - 2**
Алексей Широких, IBM

64 **IBM WebSphere Community Edition**
Enterprise-уровень начинается здесь

68 **Модуль ядра – своими руками**
Если сетевая карта не поддерживается...

72 **Ядро Linux: готовые рецепты**
А как настроить ЭТО?

76 **Учебник: первые шаги**
Групповые календари и почта

80 **Учебник Gimp**
На прощание – стеклянный эффект

84 **Учебник Inkscape**
Работаем с текстом

88 **Hardcore Linux**
Нестандартный урок по Bash

92 **Учебник PHP**
Разбираемся с представлениями

94 **Знакомство с Qt: первые шаги**
Рабочий стол изнутри

98 **Почему – VIM?**
Империя наносит ответный удар

102 **Учебник Python, часть 4**
ООП – это здорово

104 **Учебник Metapost, часть 3**
Повторение – мать учения

108 **История Cyrillic HOWTO**
Взлет и падение [еще одного] открытого проекта

110 **Linux-пресса на Руси**
От «Монитора» до LXF

114 **Вопрос? Ответ!**
FAQ по Open source

121 **Диск Linux Format**
В этом выпуске...

16 Лети, Хадсон!

38 Поговорим с Mono

На диске

Целый DVD интересных программ 124

» DVD

OPENSUSE SLICK Популярный дистрибутив, оптимизированный по скорости проектом openSUSE.

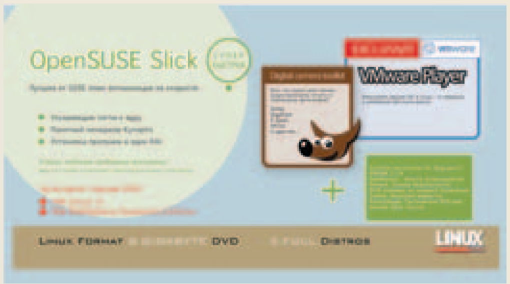
VMWARE PLAYER Запускайте Windows и другие ОС прямо в Linux

ЦИФРОВЫЕ КАМЕРЫ Полный набор приложений, упомянутых в спецрепортаже

GNOME 2.14 Настало время обновиться

GALAXYMAGE Трехмерная ролевая игра. Поупражняйтесь в тактике (и стратегии).

Вырежьте свою обложку для DVD на стр. **121-122**



Перед использованием диска ознакомьтесь с инструкцией на стр. 121

Главные новости

• Gnome 2.14 • GCC 4.1.0 • Ядро 2.6.16 • Новые дистрибутивы и антивирусы • Гаэль Дюваль уволен из Mandriva • NOD32 2.5 • Центр верификации ОС Linux

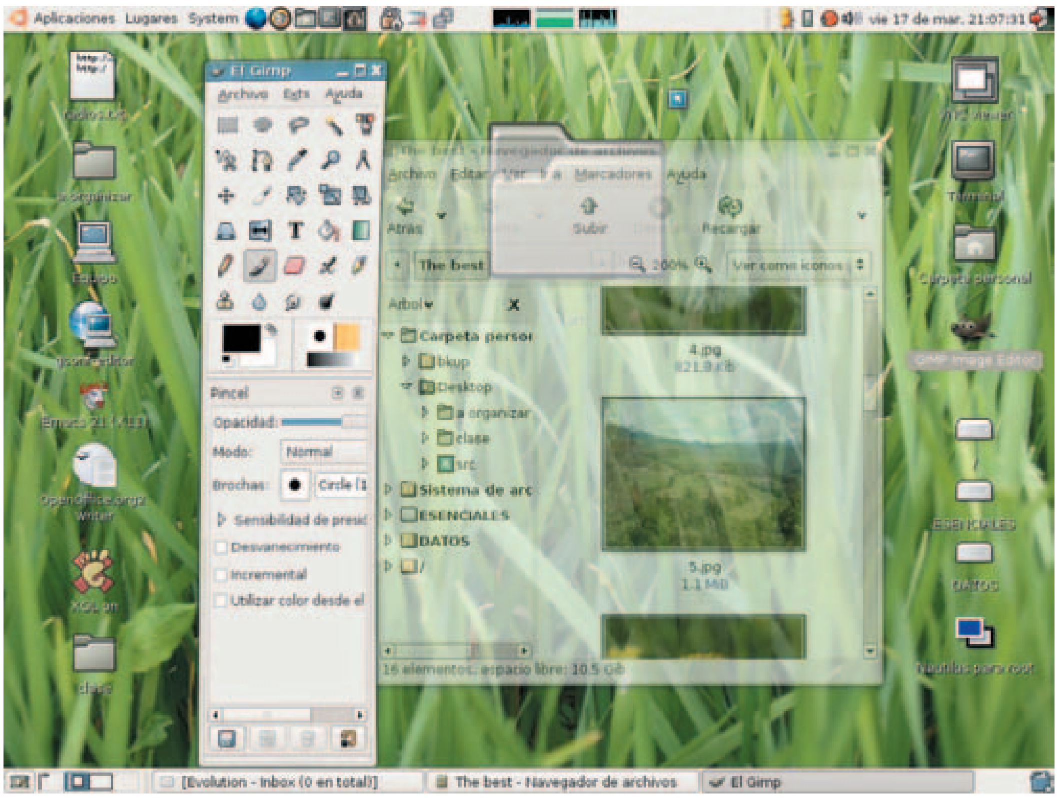
GNOME 2.14

Еще и быстрее, и с интегрированным поиском

Проект GNOME (www.gnome.org) объявил о выходе новой версии рабочего стола GNOME, лежащего в основе Ubuntu, *Fedora Core* и других дистрибутивов Linux. Как сообщают разработчики, *GNOME 2.14* стал еще более производительным – например, вывод всех английских слов в окне GNOME Terminal происходит всего за одну секунду (сравните с 3,1 сек. *GNOME 2.12* или 4,3 сек. xterm). Файл журнала, объемом без малого 3Мб, был открыт в Log Viewer за 1,8 сек., тогда как раньше для этого требовалось не менее 40. Таких впечатляющих результатов удалось добиться за счет более грамотного подхода к отрисовке экранных шрифтов и выделению памяти.

GNOME 2.14 значительно облегчает процедуру поиска. Начнем с того, что поисковый интерфейс доступен прямо в файловом менеджере *Nautilus*. При наличии Beagle, в качестве поискового движка будет использоваться именно он, что открывает поистине потрясающие возможности. Результаты поиска можно сохранять в специальную папку, чтобы вернуться к ним при необходимости. Панель поиска появилась и в браузере справочной системы, а также получила свое дальнейшее развитие в форме *Deskbar*.

Deskbar – это мини-командная строка, доступная практически отовсюду и использующая специальные расширения, написанные на *Python*. С ее помощью можно выполнять поиск по файлам, папкам, закладкам, контактом, а также в *Google Live* и *Yahoo*.



Как известно, лучше один раз увидеть, чем сто раз услышать, поэтому вы можете найти исходные тексты *GNOME 2.14* на диске этого выпуска *Linux Format*.

Если у вас есть под рукой Xgl, *GNOME 2.14* будет «просто прелестью».

GCC 4.1.0

Хит сезона – межпроцедурная оптимизация

С момента выпуска *GCC 4.0* не прошло и года, а неутомимая команда разработчиков уже представила новую версию *GNU Compilers Collection*. «Гвоздь» этого сезона – межпроцедурная оптимизация (inter-procedural optimization), которая раньше была доступна только в коммерческих продуктах навроде Intel C Compiler. Если раньше оптимизатор GCC изучал вашу программу по кусочкам (отдельным функциям), то сейчас он имеет возможность обозревать ее целиком. Естественно, работая в таких масштабах, можно увидеть куда больше, и например, выполнить константную подстановку (если какая-либо функция все время вызывается с одним и тем же набором параметров) или удалить статические переменные, которые (вследствие оптимизации) более не используются. Очевидно, что это позволит поднять скорость выполнения открытых программ (большая часть из которых собирается при помощи *GCC*) на новую высоту. Однако, как и в случае со всякой новой технологией, *GCC 4.1* не свободен от ошибок, поэтому рекомендуем не бросаться с головой в омут, а дождаться выпуска исправлений: *GCC 4.1.2 – 4.1.3*.

Впрочем, самые нетерпеливые могут использовать новые возможности уже сейчас: *GCC 4.1* включен в *Fedora Core 5* и *SUSE 10.1* (см. обзор на странице XX).

Linux 2.6.16

Кое-кто считает это началом революции

Изменения в новых версиях ядра не так сильно бросаются в глаза, как новые функции популярных рабочих столов, но именно они определяют облик мира Linux на ближайшие несколько месяцев. И здесь ядру 2.6.16 определенно есть, чем похвастаться. Мало того, что в его состав была включена кластерная файловая система *OCFS2*, разработанная компанией Oracle – Linux теперь работает на процессорах Cell, что открывает для него новые перспективы. Разработчики также реализовали несколько новых системных вызовов (**unshare()**, **pselect()/ppoll()** и ***at()**), о которых давно мечтали авторы *Glibc*, и обеспечили возможность переноса области подкачки на NUMA-системах. Появились таймеры высокого разрешения, списки контроля доступа (ACL) для файловой системы *CIFS*, а также новая файловая система *configfs*, работающая в паре с *sysfs*. Как и *GNOME 2.14*, ядро 2.6.16 можно найти на нашем диске.

Novell SUSE Linux Enterprise 10

***Xen* и *Xgl* – убойный коктейль**

Компания Novell представила SUSE Linux Enterprise 10, новую версию своего дистрибутива уровня предприятия. Это первая корпоративная платформа, включающая гипервизор *Xen 3.0* (*см. Linux Format 77*), позволяющий консолидировать множество задач на одном сервере. По данным Gartner, сервер обработки данных в среднем работает лишь на 20% своей емкости. Благодаря средствам виртуализации *Xen*, эту величину можно увеличить почти до 70%.

SUSE Linux Enterprise 10 содержит технологию AppArmor, ранее являвшуюся собственностью компании Novell, а ныне выпущенную по открытой лицензии. AppArmor помогает быстро разрабатывать правила безопасности и обеспечивать их соблюдение, защищая инфраструктуру от злоумышленников, стремящихся воспользоваться ошибками в приложениях.

Помимо перечисленных выше, *SUSE Linux Enterprise 10* содержит и другие усовершенствования, самые интересные из которых – это «ускоренная» графическая подсистема *Xgl* и специализированная версия *OpenOffice.org*, позволяющая корректно работать с макросами Microsoft Excel. Таким образом, степень совместимости открытого офисного пакета с Microsoft Office плавно приближается к 100%.

Череда дистрибутивов

Fedora Core 5, ASPLinux 11, KNOPPIX 5.0

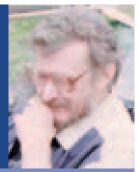
На март пришелся настоящий бум выпуска новых версий дистрибутивов. С небольшим опозданием увидела свет Fedora Core 5, включающая в себя две упомянутые выше новинки: *GCC 4.1* и GNOME 2.14. Помимо этого, следует отметить все тот же Xen (хотя для «непромышленного» дистрибутива это не такая уж новость), *Glibc 2.4, AIGLX* (альтернативу *Xgl*) и KDE 3.5.1

«Русская версия» Fedora Core – *ASPLinux*, тоже не теряла времени даром. В состав выпущенной аккурат к Международному Женскому Дню одиннадцатой версии вошли: ядро 2.6.14, GNOME 2.12, KDE 3.5.1, *OpenOffice.org 2.0.1*, *Firefox/Thunderbird 1.5*, а также фирменные драйвера NVIDIA и ATI. *ASPLinux 11*, как и его предшественники, распространяется в четырех редакциях: *Greenhorn (*LiveCD*)*, *Express*, *Standard* и *Deluxe*.

Ни и наконец, нельзя не отметить прогресс самого популярного в мире LiveCD (то есть LiveDVD) – KNOPPIX, пятая версия которого была представлена публике в ходе выставки CeBIT 2006. Помимо обновления компонентов (*ядро 2.6.15.4, X.Org 6.9, udev+hwsetup* для автоматического распознавания оборудования), стоит упомянуть прозрачную запись на разделы NTFS (libntfs+fuse) и обновление существующей инсталляции KNOPPIX на жестком диске.

На момент написания этой статьи *SUSE 10.1* еще находился в стадии финального тестирования.

Алексей Федорчук
Видел больше операционных систем, чем вы можете себе представить.



КОММЕНТАРИИ

Новый инсталлятор Debian

Debian обзавелся новым инсталлятором. Точнее, это все тот же *Debian Installer*, впервые появившийся в *Sarge*, но несколько модифицированный.

Теперь установка совершается в один этап – без перезагрузки: после развертывания базовой системы следует предложение настроить доступ к архивам пакетов, далее – выбрать целевые наборы (Standard System, Laptop, и так далее), которые немедленно и устанавливается. Так что по рестарту машины система имеет место быть в полностью скомпонованном виде.

За эту идею разработчики *Debian* выражают благодарность коллегам по Ubuntu. Хотя на самом деле они пошли дальше. Как известно, вариант *Debian* Installer от Ubuntu изначально отличался тем, что осуществлял установку «в полтора этапа» – после рестарта происходило развертывание дополнительного софта. Debian'овцам удалось целиком вписаться в один этап.

Впрочем, главное значение нового инсталлятора мне видится в другом: в продолжающемся взаимовлиянии дистрибутивов семейства *Debian* друг на друга, позволяющем сохранить внутренее единство, противостоящее тенденциям к дивергенции, господствующим в современенном дистростроении.

КОРОТКОЙ СТРОКОЙ

- Компания Xara наконец-то предоставила публичный доступ к исходным текстам **Xara LX**: **http://www.xaraxtreme.org/** Архив занимает 20Мб. Интересующиеся могут заказать SVN-доступ.
- *Linspire* и Ubuntu заключили договор о предоставлении доступа к репозиторию **CNR Warehouse** для пользователей Ubuntu.
- Увидела свет новая версия ***OpenOffice.org – 2.0.2***. Обещается поддержка дополнительных форматов и улучшенная интеграция с KDE и Gnome.
- Под кодовым названием **Bon Echo Alpha 1** была выпущена первая альфа-версия *Firefox 2*.
- Компания Red Hat предоставила свободный доступ к базе знаний **Red Hat Knowledgebase**.

ЧТО БЫЛО

- **Юбилейный 10-й Российский Интернет Форум РИФ-2006**

26 марта в пансионате «Лесные дали» завершил свою работу Юбилейный Десятый Российский Интернет Форум (РИФ-2006), организатором которого выступил Региональный Общественный Центр Интернет Технологий (РОЦИТ).

В этом году РИФ проходил под патронатом «Федерального агентства по печати и массовым коммуникациям» при поддержке Федерального агентства по информационным технологиям и Федерального агентства по образованию.

Впервые за всю историю РИФ общее количество участников Форума превысило 1000 человек: участниками РИФ-2006 стали 1327 человек из более чем 800 организаций. Особое внимание в ходе РИФ-2006 было уделено развитию информационных технологий в российских регионах. Так, впервые в истории Форума был организован специальный региональный блок «Дни регионов РФ», главной задачей которого стало формирование карты Всероссийского РИФ-Марафона, старт которого намечен на апрель 2006 года. «Всероссийский РИФ-Марафон» представляет собой серию выездных семинаров РОЦИТ, призванных обсудить самые острые проблемы Рунета и региональные аспекты его развития.

Гаэль Дюваль уволен из Mandriva

Спустя семь лет после запуска проекта Mandrake Linux, его основатель остался не у дел

Информация, которая зародилась как слух в одном из форумов *Mandriva Club*, получила официальное подтверждение. Гаэль Дюваль (Gaël Duval), основатель проекта Mandrake Linux, интервью с которым было опубликовано в *LXF70*, уволен из Mandriva. Точнее, как поправил журналистов исполнительный директор Франсуа Бансильон (Francois Bancilhon), «попал под сокращение». То же самое говорит и сам Гаэль на специально открытой «по случаю» web-странице (**http://www.indidea.org/gael/en/fired-message.php**): «Уволен. Да, просто уволен, по экономическим соображениям, вместе с некоторыми другими. Спустя семь лет после того, как я создал Mandrake-Linux и, затем, Mandrakesoft, текущий босс Mandriva поблагодарил меня и выставил за дверь со стандартной компенсацией в объеме двух месячных окладов. Это трудно принять: семь лет назад я создал рабочие места для себя и многих других, а в прекрасный февральский полдень исполнительный директор Mandriva позвонил мне и сказал, что я сокращен».

Формальным поводом для увольнения г-на Дюваля послужило стремление компании сократить свои расходы, однако, как отмечает сам Жиль, не последнюю роль могли сыграть и его натянутые отношения с Франсуа Бансильоном. Основатель Mandrakesoft подал на компанию в суд за неправомочное увольнение, однако, судя по всему, не собирается возвращаться. Вместо этого, он сфокусирует свои усилия на Ulteo (**www.ulteo.com**) – открытой ОС нового поколения. На момент написания данной статьи на сайте работал только список рассылки, однако, есть надежда, что к тому времени, когда вы будете читать эти строки, выйдет в свет первая бета-версия Ulteo. Вот тогда и разберемся.

NOD32 2.5

Новая версия антивируса для Linux

Сравнение антивирусных решений для Linux было опубликовано в январском номере *LXF*, а в этом выпуске вы найдете обзор Sophos Anti-Virus. Но жизнь не стоит на месте и сегодня мы рады сообщить вам о выходе новой версии антивируса *NOD32* от компании Eset.

При создании *NOD32 v2.5* для серверов Linux основные усилия разработчиков были сосредоточены на повышении стабильности, эффективности и настраиваемости для конкретных категорий пользователей. Теперь основной системный процесс *NOD32* выполняет исключительно надзорные функции, запуская по мере необходимости нужное число подчиненных процессов, которые и выполняют работу по выявлению и обезвреживанию вирусов в конкретных информационных объектах – файлах, структурных частях электронного сообщения и т.п. Помимо этого, *NOD32 v2.5* позволяет указать максимальную глубину сканирования вложенных архивов и обладает доработанной функцией карантина.

Узнать больше об этом решении и загрузить пробную версию можно на сайте компании: **www.esetnod32.ru**

В России открылся Центр верификации ОС Linux

Формализация стандартов Linux и разработка открытого тестового набора была поручена ИСП РАН

Институт системного программирования Российской Академии Наук (ИСП РАН) выиграл конкурс Федерального агентства по науке и инновациям на формализацию стандартов для Linux, разработку соответствующего открытого тестового набора и создание Центра верификации ОС Linux, **linuxtesting.ru**. В рамках конкурса предполагалось формализовать стандарт *LSB 3.1* и создать на его основе набор тестов для ОС Linux. Эти тесты будут находиться в открытом доступе, а для их поддержки и развития и был организован Центр верификации ОС Linux, который будет действовать на базе Института системного программирования Российской Академии Наук (ИСП РАН).

Директор ИСП РАН, член-корреспондент РАН Виктор Петрович Иванников на совещании по Open Source в России сказал: «ИСП РАН рассматривает этот проект как старт долгосрочной программы, нацеленной на повышение добротности системного программного обеспечения в мире Linux. Спецификация интерфейсов операционной системы и их стандартизация - это первый шаг в создании надежных, совместимых платформ для различных приложений». LXF

Обзоры >>

Новинки программного и аппаратного обеспечения с точки зрения наших экспертов.

Процессор Cell и его роль в Linux-революции

Алексей Федорчук
Написал эту колонку в Joe.

Много лет назад (1992 г.) в журнале PC Magazine появилась статья под зловещим названием: «Через десять лет все платформы, кроме IBM PC, уйдут в небытие». Тогда это казалось невероятным.

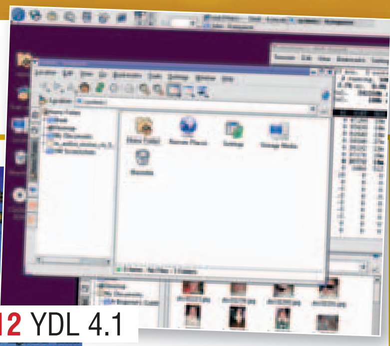
Увы – мрачный прогноз сбывался опережающими темпами. И на рабочих столах пользователей безраздельно воцарились процессоры от Intel, все прочие архитектуры ушли в нишевые области. А недавно рухнул последний бастион антиинтеловской обороны: Apple начала переход на PC-архитектуру.

Конечно, стандартизация оборудования имела свои положительные стороны. Однако согласитесь – в унифицированном мире жить просто скучно...

И вот – процессоры Cell от фирмы IBM. Революционные архитектурно, фантастически производительные и сверхъестественно дешевые. Могущие в равной мере служить сердцем и игровых станций, и сверхмощных серверов. Которым не хватает лишь одного – адекватной ОС. Каковая не замедлила появиться: в ядре Linux версии 2.6.16 заявлена поддержка процессоров Cell. И этот факт может повернуть все ее развитие.

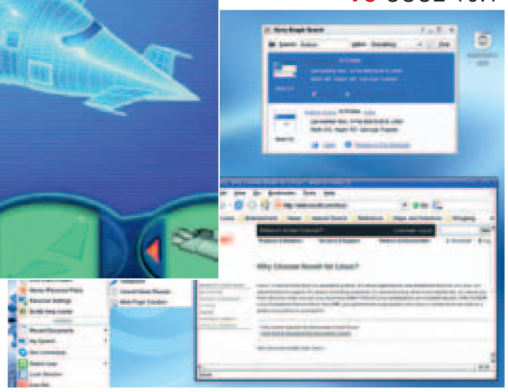
Представим себе игровые станции на процессорах Cell под Linux. Это даст стимул к совершенствованию графической подсистемы, что приведет наконец к пресловутой ее «десктопизации». Рискну предположить, что «десктопизация» Linux пойдет совсем не так, как DOS и Windows – не к тотальному универсализму, а к специализированным станциям (пусть сначала и игровым). То есть – генеральным путем развития Unix и Linux.

ЧТО НОВЕНЬКОГО?



12 YDL 4.1

16 Airline Tycoon



10 SUSE 10.1

10 SUSE 10.1
С момента выпуска SUSE 10.0 прошло шесть месяцев, так что это обновление как раз кстати.

12 YELLOW DOG LINUX 4.1
Apple переходит на Intel, YDL остается при своих. Теперь – летающей тарелкой!

14 GENTOO 2006.0
Да, он по прежнему собирается из исходников. И это по-прежнему долго. Так что же изменилось?

15 ARKEIA SMART BACKUP
Коммерческая система резервирования по новым ценам. Удастся ли сэкономить?

16 AIRLINE TYCOON
Развлекитесь: управляйте собственной авиакомпанией при помощи опыта, денег и мелких подлянок.

НАШ ВЕРДИКТ: ПОЯСНЕНИЕ

Все продукты оцениваются по 10-бальной шкале. 4 обычных параметра оценки: возможности, производительность, простота использования и соотношение «цена/качество», но для свободного ПО последний параметр может быть заменен на оценку документации. Независимо от набора категорий, мы всегда вычисляем общий рейтинг, подводящий итог нашим высказываниям.

Продукты, выделяющиеся из основной массы, получают престижную награду *Linux Format Top Stuff Award*. Выбираются самые-самые – и только лишь высокая оценка здесь еще ни о чем не говорит.

Для тестирования серверов используется наша собственная разработка, LFXBench 2004, состоящая из 4 основных подсистем: Multi CPU, Single CPU, RAM и Hard Disk. Оценки усредняются и дают общий рейтинг.

1 очко означает, что тестовый компьютер равен нашему эталону: Pentium 4 1,8ГГц, 512 Мб оперативной памяти и жесткий диск IDE. 2 очка означает, что он вдвое быстрее.

Все тесты выполняются под управлением Red Hat Enterprise Linux 3 AS на соответствующей платформе (x86, AMD64, Itanium). Код компилируется при помощи GCC, если не указано иное.

РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ

MULTI CPU	6.07
SINGLE CPU	3
RAM	2.17
HARD DISK	0.46
ОБЩИЙ РЕЙТИНГ	2.93

ДИСТРИБУТИВ LINUX

SUSE 10.1

SUSE выпустила новый дистрибутив – точно, как часы Black Forest. **Майк Сондерс** его опробовал.

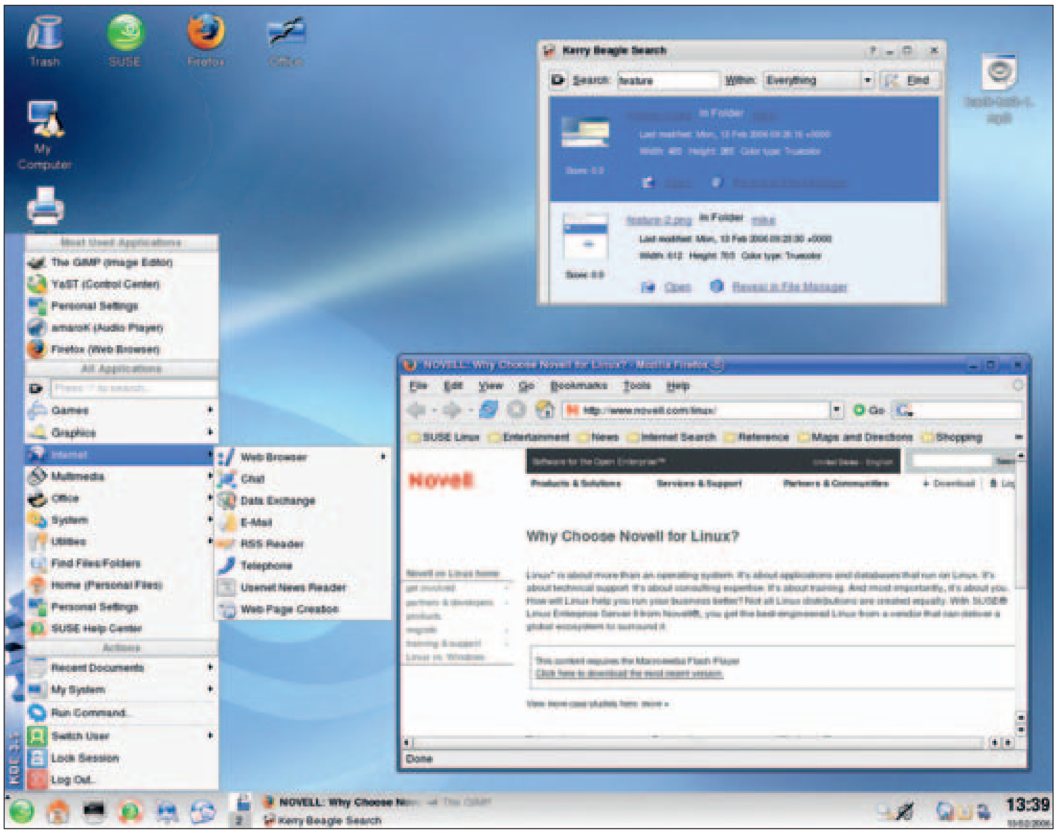
САМОЕ ГЛАВНОЕ

RPM-дистрибутив для дома и малых офисов. Альтернативы: Mandriva, Ubuntu или Fedora Core.

- **РАЗРАБОТЧИК:** Novell
- **САЙТ:** www.novell.com
- **ЦЕНА:** Свободная загрузка или коробочная версия, цена согласно описанию

Если что-то в мире Linux и гарантировано, то это стабильность выпуска релизов SUSE. Невзирая на перемены, недавно сотрясавшие команду разработчиков – например, переход в собственность Novell с последовавшим созданием проекта OpenSUSE – они сохраняют жесткую периодичность шлифовки и отгрузки дистрибутива. Иногда SUSE идет на риск, включая новейшие версии программ, но постоянная забота команды инженеров и регулярные обновления вселяют уверенность, что стабильность дистрибутива очень быстро будет выведена из-под удара.

SUSE Linux 10.1 поставляется на CD; в случае загрузки ISO-образов вы можете преобразовать их в DVD, используя **http://en.opensuse.org/Making_a_DVD_from_CDs**. Дистрибутив требует минимум 64 МБ ОЗУ для установки в текстовом режиме или 128 МБ для графического. Подойдет любой процессор чуть получше Pentium'а, но графический режим полноценно заработает на частоте от 1 ГГц. Требования к свободному месту на жестком диске зависят от выбора устанавливаемых приложений, но мы советуем припасти не менее 3 ГБ.



Novell создала почти идентичные темы для приложений *GTK* и *Qt/KDE*, придав блеск рабочему столу.

Оцените...

Процесс установки почти не изменился: все те же прекрасные панели со справочным текстом и превосходное обнаружение устройств. На выбор предлагаются Gnome и KDE, установки по умолчанию нет, и пользователь должен принять решение сам.

Огорчает только одно: временами инсталлятор задумывается, особенно когда копирует приложения.

> «ИЗРЯДНО ПОХУДЕВ, НАСТОЛЬНЫЙ SUSE СЕЛ НА ХВОСТ UBUNTU.»

С загрузкой система тоже не спешит, но ее шарм трудно игнорировать: у суперкорпоративных окон входа в систему и загрузки KDE – располагающе-профессиональный вид, и на каждом этапе заметно влияние Novell. Так, *OpenOffice.org* и *Gimp* снабжены стартовыми окнами с логотипом Novell, придающими целостность дистрибутиву. Вы ощущаете, что получили не универсамовскую корзину с программами, а оболочку ручной работы.

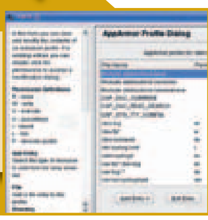
Ощущение целостности присутствует и в K-меню для KDE. Как в 10.0, SUSE предпочла обойтись без жутких лабиринтов подменю и 15 текстовых редакторов на выбор. Вместо этого почти в каждом пункте присутствует установка по умолчанию всего одной программы, названной по выполняемым функциям, а не по прихоти программиста. Например, *Gimp* назван 'Редактор изображений', а Kopete – 'Обмен сообщениями'. Версия 10.1 еще улучшилась: меню более строго упорядочено, а названия его пунктов более информативны.

Новые радости

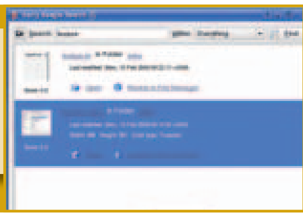
«Под капотом» был удален *hotplug* – его функции взял на себя *udev*. SUSE 10.1 не выглядит абсолютно новым рабочим столом в смысле программ, но есть несколько примечательных дополнений. Наиболее важно и заметно для пользователя обновление KDE до версии 3.5, с новым меха-

ОДНИМ ВЗГЛЯДОМ

***AppArmor* 2.0, система безопасности от Novell, доступна через Yast.**



***Kerry* добавил к среде KDE модный инструмент *Beagle*.**



***NetworkManager* упростил переключение сетей в 10.1.**

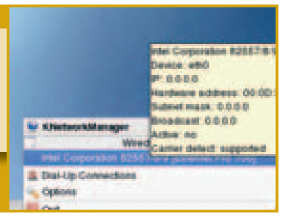


ТАБЛИЦА ДЛЯ СРАВНЕНИЯ SUSE 10.0 И 10.1

	10.0	10.1
Ядро	2.6.13	2.6.16
Gnome	2.12.0	2.12.2
KDE	3.4.2	3.5.1
Apache	2.0.54	2.2.0
Firefox	1.0.6	1.5.0.1
GCC	4.0.2	4.1.0
Gimp	2.2.8	2.2.10
MySQL	4.1.13	5.0.18
OpenOffice.org	2.0	2.0.1
X.org	6.8.2	6.9.0
Размер установки*	2.0Гб	2.1Гб
Использование ОЗУ**	95Мб	120Мб
Время загрузки	55 сек	100 сек
Время выключения	24 сек	32 сек

* Для установки по умолчанию
** Стандартное значение для KDE после загрузки.
Проверено на: 2.4GHz Celeron ПК с 256 МБ ОЗУ

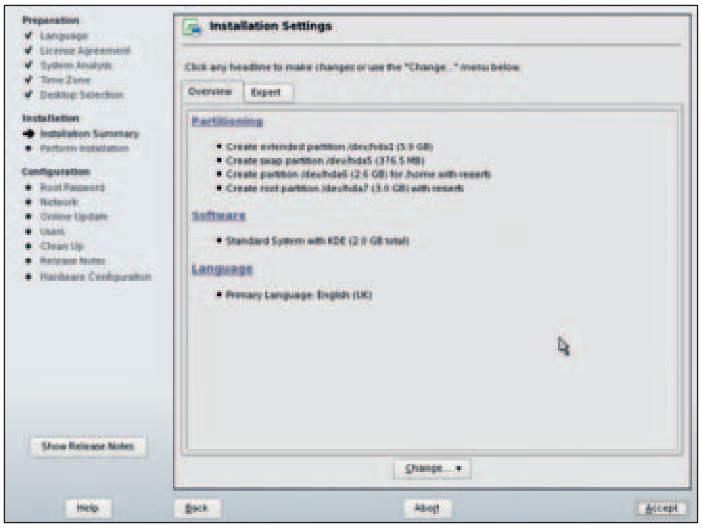
низмом блокировки всплывающих окон в *Konqueror*, встроенной *SuperKaramba* и поддержкой web-камер для MSN/Yahoo. Firefox прыгнул от версии 1.0 к 1.5, Xen обновился до 3.0, Mono до 1.1.3, а вот Gnome и *OpenOffice.org* не особо изменились. По юридическим причинам, воспроизведение MP3 и DVD в бесплатной версии по умолчанию недоступно.

Kerry добавил к окружению KDE Beagle, движок для поиска, по которому чуть ли не год все сходили с ума. Это весьма минималистичная оболочка, и она выполняет свою работу, не забивая память библиотеками Gnome. Новое приложение *NetworkManager*, от команды Red Hat, предназначено для беспроблемного использования Linux-машины в сети. Оно отображается как аплет KDE в системном лотке и позволяет плавно переключаться между сетями. Это не образец совершенства, но удобнее, чем мучить *Yast* при каждом переключении от одной сети к другой или настройке нового подключения.

Последнее важное новое приложение – *AppArmor 2.0*. Изначально это был закрытый проект Novell, но в январе он переведен в разряд Open Source и теперь входит в стандартный набор SUSE 10.1. *AppArmor* представляет среду обнаружения и защиты от атак, позволяет определять права для программ и отслеживать подозрительную активность; во многом оно похоже на *SELinux* и довольно легко конфигурируется через *Yast*.

И как всегда, в наличии *Yast* – всеобъемлющая утилита конфигурирования. Многие пользователи SUSE и любят и ненавидят *Yast*. Любят за исчерпывающий набор инструментов конфигурирования, мастеров и диалоговых окон, охватывающих практически всю систему; ненавидят за способность затоптать ручные изменения в файлах конфигурации и кучу скриптов поддержки, которые выполняются годами. В сети жарко спорят, лучше ли он набора отдельных утилит (как в Fedora или Ubuntu) или нет.

Yast все еще не столь дружественен, как Центр Управления Mandriva (т.к. предназначен скорее для администраторов, чем для пользователей), но выполняет впечатляющую работу по управлению и настройке всех опций системы. Установка программ, конфигурирование оборудования, настройка межсетевого экрана – тут есть все, вдобавок также и в текстовом режиме. В идеале, нам бы хотелось видеть в Yast режим «попроще» – возможно, с другой оболочкой и подсказками, чтобы не отпугивать новичков.



Установка почти не изменилась, она по-прежнему дружественная и превосходно определяет оборудование.

Конкуренция

В обзоре SUSE 10.0 в *LXF4(73)* мы отмечали, что SUSE скоро станет ближайшим соперником Ubuntu на рабочих столах. С ростом популярности Ubuntu у конкурирующих дистрибутивов появляется все больше причин его копировать, и команде SUSE следует грамотно обдумать, куда приложить усилия: добавлять в изобилии новые пакеты на диски или создать очень простой рабочий стол с выбором единственного приложения и чрезвычайно устойчивым GUI. Такого направления придерживается Ubuntu, и его популярность подтверждает верность данного выбора.

К счастью, теперь разработчики SUSE осознали достоинства подобного дизайна – урезанный набор стандартных приложений и интуитивно-понятные имена пунктов меню в новом релизе – тому свидетельства. У SUSE есть свой фан-клуб продвинутых пользователей, которые привыкли к массе протестированных приложений в одной удобной вязанке. Естественно, SUSE не хочет терять старых клиентов, но уменьшая размер установки по умолчанию и поддерживая общий репозитарий пакетов, он садится на хвост к Ubuntu. В ближайшую пару лет ждите волнующих преобразований структуры рынка настольных дистрибутивов.

Итак, если вы пользователь SUSE 10.0, стоит ли обновляться? В релизе нет ничего ошеломляюще-нового, хотя изначальная интеграция с *NetworkManager* и обновленные версии Gnome и KDE могут подвигнуть на переход охотников за свежими версиями приложений. Скорость загрузки позорно медленная; может быть, доработка SUPER (*см. врезку Открывая новые двери, слева внизу*) поправит дело в следующем релизе. Тем, кто не чувствует восторга, лучше подождать следующего «мажорного» релиза.

ГРЭМ СЧИТАЕТ…

«Мне нравится подача данного релиза. Но, по-моему, для многих пользователей изменений маловато, чтобы перейти с хорошо сконфигурированного SUSE 10.»

И наоборот, если вы новичок в Linux (или хотите сменить дистрибутив), то SUSE 10.1 – прекрасный выбор: здесь превосходный рабочий стол, изобилие программного обеспечения и бойкое сообщество. Отлично сделано, Novell: хорошо отшлифованный дистрибутив, не уронивший привычно-высокий стандарт. LXF

ОТКРЫВАЯ НОВЫЕ ДВЕРИ

В августе 2005 Novell заявила о решении сделать разработку SUSE более прозрачной и доступной под именем OpenSUSE – попытка привлечь к себе более широкое сообщество, аналогично Red Hat с Fedora Core. Однако Novell очень быстро подчеркнула, что OpenSUSE не просто обкаточная бета-версия для стабилизации коммерческого продукта, но полностью разработанный и протестированный дистрибутив для домашних пользователей.

Мы поговорили с Грегом Манкузи-Унгаро (Greg Mancusi-Ungaro), сообщившим: «С учетом разработчиков, дизайнеров интерфейса пользователя, экспертов по документации и конечных пользователей, уже зарегистрировано 20 000 членов сообщества **OpenSUSE.org**. Сообщения об ошибках положительно сказались на качестве дистрибутива SUSE Linux.» Также, согласно Манкузи-Унгаро, число подтвержденных загрузок SUSE 10.0 уже достигло миллиона. «Пользователи расхватывали продукт со скоростью 7000 установок в день. Копия SUSE Linux устанавливалась почти каждые 12 секунд.»

Пока что проект OpenSUSE предлагал пользователям доступ к обычным альфа- и бета-версиям с возможностью их загрузки для тестирования и сообщения об ошибках разработчикам. Теперь график OpenSUSE приблизился к Стадии 3, подразумевающей создание полной обобщенной инфраструктуры, где сторонние участники могут добавить собственные пакеты в тестовый выпуск дистрибутива. Имеются также побочные ветви, типа SLICK и SUPER – выпуски с улучшенной производительностью и функциональностью (дополнительную информацию смотри в описании диска-приложения с.106)

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	9/10
ПРОИЗВОДИТЕЛЬНОСТЬ	7/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	8/10
ДОКУМЕНТАЦИЯ	9/10

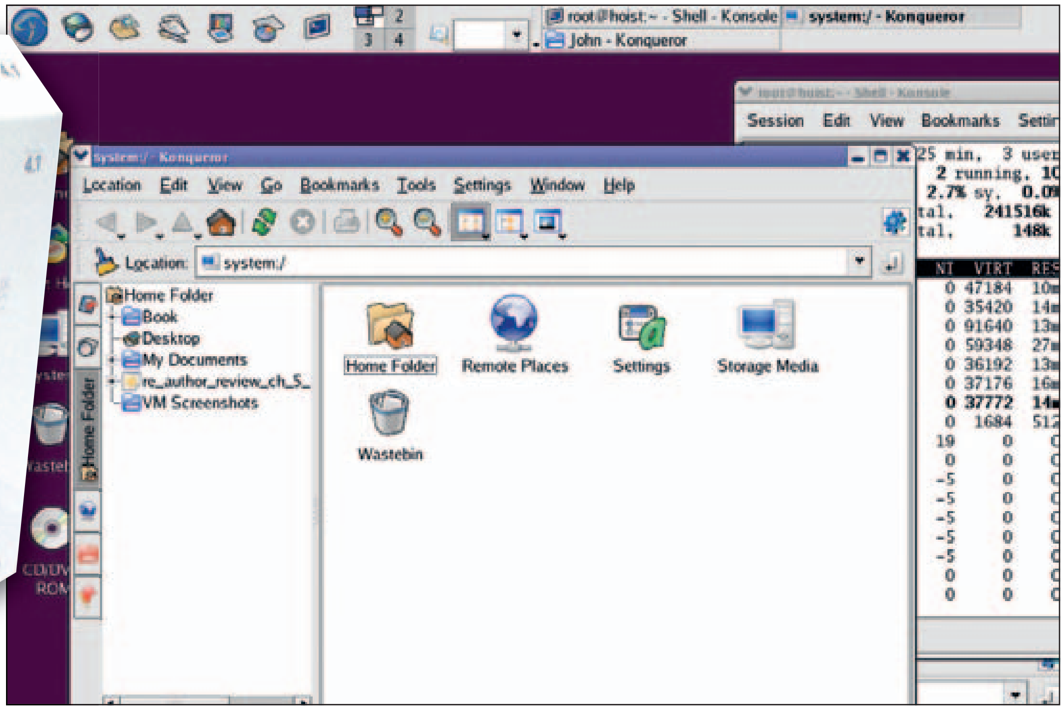
Пользователей версии 10.0 соблазнить на обновление особо нечем, но всетаки это неплохо поданный дистрибутив: и функциональный, и дружелюбный к новичкам.

РЕЙТИНГ 8/10

ДИСТРИБУТИВ LINUX

Yellow Dog Linux 4.1

Зажатый между новой архитектурой Apple с одной стороны и Linux-конкурентами на PowerPC – с другой, дистрибутив Yellow Dog нуждается в защите своего места под солнцем. **Энди Хадсон** запустил его на своем доинтеловском Mac'е.



Это KDE, Джим, но не тот, каким ты знаешь – Yellow Dog Linux продвинула Bluecurve еще на шаг.

САМОЕ ГЛАВНОЕ

Дистрибутив для PPC, настроенный на оборудование Apple. Альтернативы: Fedora Core 5 (PPC) или Kubuntu PPC.

- **РАЗРАБОТЧИК:** Terra Soft Solutions
- **САЙТ:** www.terrasoftsolutions.com
- **ЦЕНА:** $89.99 или $59.99 без поддержки

Спутник на всю жизнь? Хозяева надеются, что у их любимцев в запасе достаточно собачьих лет, но если говорить о Yellow Dog (Желтой Собаке) – его владельцы часто сбегают быстрее, чем собаки наших дней, особенно потому, что Apple больше не комплектуются чипами PowerPC. И все-таки Yellow Dog долго был верным дистрибутивом для многих пользователей Linux, и нам очень интересно рассмотреть работу последней версии в качестве замены Mac OS X.

За семь лет Terra Soft Solutions создала для себя нишу на рынке архитектуры Power, продавая YDL. То, что она является реселлером Apple, выделяет ее как эксперта в данной области, придавая уверенность тем, кто сталкивается с YDL впервые. Выпущенный в самом начале 2006, YDL 4.1 является первым обновлением за год для Power-специалистов. Оправдан ли этот срок?

СИСТЕМНЫЕ ТРЕБОВАНИЯ

YDL – дистрибутив для PowerPC, и ему требуются машины на основе Power, включая G3–G5 PowerMac, G3–G4 iBook, G3–G5 iMac, G3–G4 PowerBook, Mac Mini и Xserves. *Mac-on-Linux* не будет работать ни с G5-системами, ни с графическими картами Nvidia. Новые iMac и MacBook Pro основаны на Intel, и YDL на них работать не будет.

Ко мне, парень!

Вы можете получить YDL одним из трех способов: купить его в коробочном варианте через Интернет (**www.terrasoftsolutions.com/store**); присоединиться к сети YDL.net – многоярусной системе подписчиков, предоставляющей доступ к ISO-образам для загрузки; или, если терпения хватит, использовать публичные зеркала. Все покупки в онлайн-магазине Terra Soft стоимостью свыше $90 предоставляют пользователям автоматический доступ к основному уровню YDL.net, но вам необходимо будет продвинуться до уровня Enhanced, чтобы получить доступ к свежим ISO выпускающихся релизов.

Для полноты ощущений мы купили коробочную версию, и рекомендуем всем новичкам в Linux сделать то же самое. Коробочные версии в наши дни становятся раритетом, главным образом из-за высокой сменяемости дистрибутивов и расходов на доставку, поэтому просто здорово, что Terra Soft может поставлять свой дистрибутив в таком виде. Вы получаете не только 8 CD с дистрибутивом и исходными текстами, но и клевую летающую тарелку, наклейки, а также приятное и полезное 182-страничное руководство по использованию YDL 4.1 (спешим сообщить, что летающая тарелка действует прекрасно: мы ею в офисе покидались). Последним извлекли из коробки руководство по установке Yellow Dog Linux 4.0.1. Номер, конечно, слегка разочаровывает, потому как остальные части пакета все-таки изменились.

Переход на Yellow Dog для нового пользователя – шаг серьезный. Установка дистрибутива Linux для PPC требует полного уничтожения раздела Mac OS X. Самый легкий способ сделать это – использовать дисковую утилиту с установочного диска Mac OS X для создания двух разделов на вашем Mac'е.

YDL: ОСНОВНЫЕ ПРОГРАММЫ

- **Ядро 2.6.15**
- **Gnome 2.12.1**
- **KDE 3.4.2**
- ***Apache 2.0***
- ***Evolution 2.2***
- ***Firefox 1.5***
- ***Mac-on-Linux 0.9.71***
- ***MySQL 3.23.56***
- ***OpenOffice.org 1.1.2***
- ***X.org 6.8***

Когда вы разделаетесь с разделами, установка пойдет гладко и займет около 20 минут до полной готовности Workstation (Рабочей Станции) стандартного типа. Если вы используете Fedora, то вам будет знаком установщик YDL, основанный на *Anaconda*. После перезагрузки вы попадаете в начальную конфигурацию первой загрузки, а затем в окно входа.

Кажется странным, что взяв за основу YDL Fedora и Red Hat Linux, Terra Soft выбрала KDE для использования в качестве стандартного окружения, хотя панели рабочего стола с выпадающими Y-меню напоминают

рабочий стол Mac. Terra Soft проделала большую работу по стиранию различий между KDE и Gnome, до такой степени, что первым делом хочется спросить «где я?». Вы свободно можете изменить настройки по вашему вкусу через Центр Управления и вернуть милую сердцу традиционную обстановку, ну, если вам это нужно.

Хотя фанатов приведет в ужас мысль, что KDE и Gnome выглядят одинаково, существует некоторая разница между реорганизацией внешнего вида и поведением. Новые пользователи будут чувствовать себя комфортно в предлагаемом меню. По нашему мнению, KDE иногда страдает от чрезмерно большого выбора возможностей, которое может обескуражить новообращенных. А тут – при минимализме, характерном для Gnome, приятно удивляет KDE-подобное богатство ассортимента.

Пестрая смесь

Теперь займемся стандартным списком пакетов Yellow Dog. С виду все то, что и полагается: ядро 2.6.15, KDE 3.4 и Gnome 1.12, а также стандартный набор разнородных приложений, окружающих базовую систему. И как-то неожиданно, что Terra Soft засиделась на *OpenOffice.org 1.1.2*, более чем устаревшей версии, которая, что еще более странно, соседствует со свежими релизами *Firefox 1.5* и *Thunderbird 1.5*.

Углубившись в серверное ПО, мы нашли достаточно эклектичную смесь *Samba 3* и *Apache 2.0* бок о бок с PHP 4.3 и *MySQL 3.23*. Сравнивая YDL с грядущей Fedora 5.0, невозможно удержаться от удивления: как это Fedora ухитрилась включить не только *MySQL 4.1*, но даже PHP 5.

Ввиду большой задержки в выпуске YDL 4.1, не удивлюсь, если вы спросите, почему это всего лишь инкрементный релиз. Кай Стаатс из Terra Soft рассказал нам, что 4.1 должен был выйти в середине 2005 года, но во время разработки обнаружились довольно серьезные ошибки, связанные с GCC 3.3. Так как YDL в то время сильно зависел от 3.3, было принято решение пересобрать все на 3.4, а когда Стаатс говорит «*все*», он имеет в виду все.

Зато дополнительное время дало возможность Terra Soft приблизиться к идеалу Apple: когда все «просто работает». Команда гордо заявила, что поддержка аудио «работает, и всё»; и она-таки работает, а в YDL 4.0 этого не наблюдалось. Для коробочной версии доступна поддержка Dual-head, тогда как поддержка теплоотвода современного G5 и беспроводных карт на основе вездесущего AirPort Extreme/Broadcom находится в разработке. Такой тип интеграции действительно делает Yellow Dog лидером, хотя другие дистрибутивы начинают его догонять.

Переходя на YDL

Одним из выдающихся шагов YDL 4.1 является включение *Mac-on-Linux*, приложения, позволяющего вам запустить Mac OS X в качестве виртуальной машины, практически не теряя в производительности. Похоже, она работает почти без проблем, даже предоставляет доступ к разделам Mac OS X Tiger на Mac Mini, подключенном через FireWire. Скорость весьма впечатляет, и мы смогли запустить все наши программы.

Однако главный вопрос – какая от этого реальная польза. Если вы не член команды разработчиков со спецзаданием на запуск Mac OS X как виртуальной машины, то вы с трудом придумаете, где использовать *Mac-on-Linux*. Это скорее некая новинка, а не насущная необходимость.

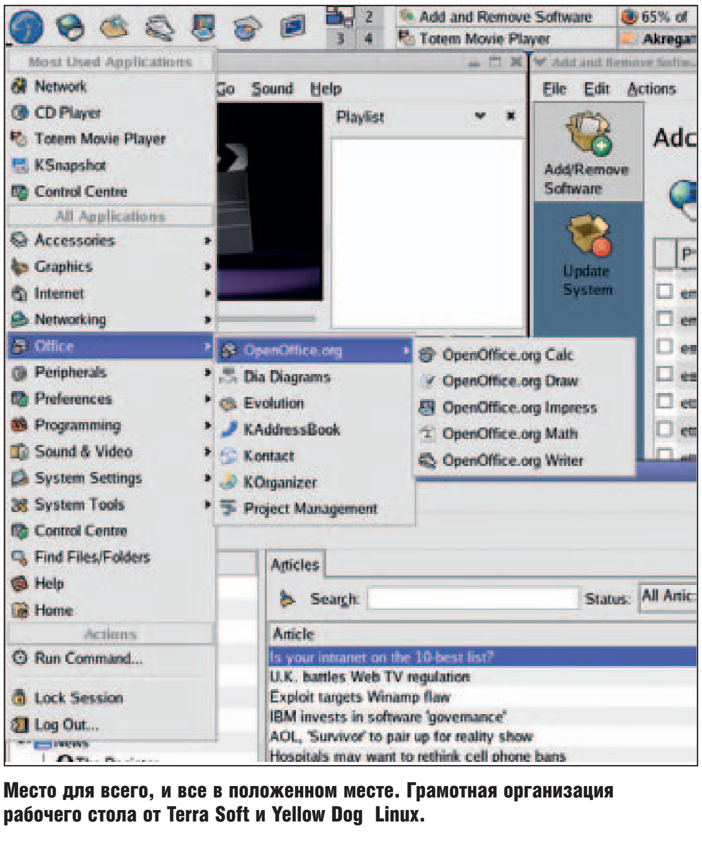
Тут появляются трудные вопросы к дистрибутиву в целом. Два года назад YDL был одним из немногих игроков в мире Linux PPC, чьи качество и редкость делали его выбор неизбежным. Теперь, когда дистрибутивы-тяжеловесы Fedora, Ubuntu и даже OpenSUSE выпустили свои PPC-версии, следует задуматься, способен ли YDL на конкуренцию. Из-за этого, да еще в сочетании с текущим переключением Apple на архитектуру x86, спрос на Yellow Dog может снизиться, по крайней мере на рабочих столах.

Сильная поддержка архитектуры фирмой IBM, особенно в ее конфигурациях OpenPOWER и BladeCenter, возможно, сигнал для YDL перейти в область серверов. Естественно, другой потенциально важной архитектурой, имеющейся на горизонте, является процессор Cell, и когда мы спросили, не это ли направление выберет Yellow Dog, Стаатс констатировал, что компания «конечно, весьма интересуется возможностями Cell на многих аренах». Заявление Sony о желании поставлять PS3 с прилагающимся дистрибутивом Linux, возможно, откроет новый массовый рынок для Yellow Dog.

Вернемся к рабочему столу: надо ли брать Yellow Dog для нашего Mac'а? Вообще-то это зависит от многого. Если вы искали дистрибутив для PowerPC, основанный на KDE и имеющий прекрасную поддержку оборудования Apple, тогда весьма возможно. Terra Soft предоставляет прекрасную платную поддержку YDL, с быстрым временем отклика, и, являясь создателем дистрибутива, может быстро менять содержимое CD для заказчиков.

Но нельзя не отметить наличие нескольких устаревших пакетов. Fedora 5.0 выглядит как целостный релиз, исправивший даже проблемы драйверов (включая ненавистный AirPort Extreme), так что сначала как-то хочется посмотреть Fedora, а уж потом обратиться к YDL.

Порох в пороховницах, тем не менее, есть. Кай Стаатс сказал, что YDL 5.0 «будет на равных конкурировать с любой приличной ОС на планете» и что Terra Soft вновь переносит свое внимание на настольные компьютеры – значит, фирма Terra Soft верит, что Power останется надолго. **LXF**



Место для всего, и все в положенном месте. Грамотная организация рабочего стола от Terra Soft и Yellow Dog Linux.

ЖИТЬ ЛИ ЖЕЛТОЙ СОБАКЕ?

Мы спросили Кая Стаатса (Kai Staats), исполнительного директора Terra Soft Solutions: справятся ли они?

LXF: После того, как Apple покинула Power-рынок, где теперь место Yellow Dog Linux среди других дистрибутивов, переключившихся на архитектуру Power (допустим, Ubuntu, Fedora и OpenSUSE)?

KC: За последние шесть месяцев мы получили больше OEM-контрактов, чем за последние шесть лет. Интерес к Yellow Dog Linux и Y-HPC для не Apple Power-устройств быстро растет, и мы горды тем, что находимся в центре этого роста. Что до других дистрибутивов, возвращающихся на PowerPC, я нахожу этот период интересным по контрасту с уходом Apple. Кажется, они немного опоздали на поезд Apple!

Праведные усилия сообщества весьма полезны и необходимы для королевства Open Source. Они альфа и омега ОС с открытым исходным кодом. Но кто-то где-то должен предлагать сопутствующую техническую поддержку и сервисное обслуживание для операционной системы. Вот применение нашим знаниям и опыту.

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	6/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	8/10
ЦЕНА/КАЧЕСТВО	7/10

Эклектическая смесь пакетов только помогает выделиться конкурентам. Если вы поклонник YDL, оставайтесь с ним, в противном случае изучайте Fedora PPC.

РЕЙТИНГ 7/10

ДИСТРИБУТИВ LINUX

Gentoo Linux 2006.0

Изменив процесс установки, потерял ли Gentoo свою бирку «ручная работа»? **Нейл Ботвик** разберется.

САМОЕ ГЛАВНОЕ

Дистрибутив на основе исходных текстов, под полным контролем пользователя.
См. также: Linux From Scratch..

- **РАЗРАБОТЧИК:** Gentoo Foundation
- **САЙТ:** www.gentoo.org
- **ЦЕНА:** Бесплатно по лицензии GPL

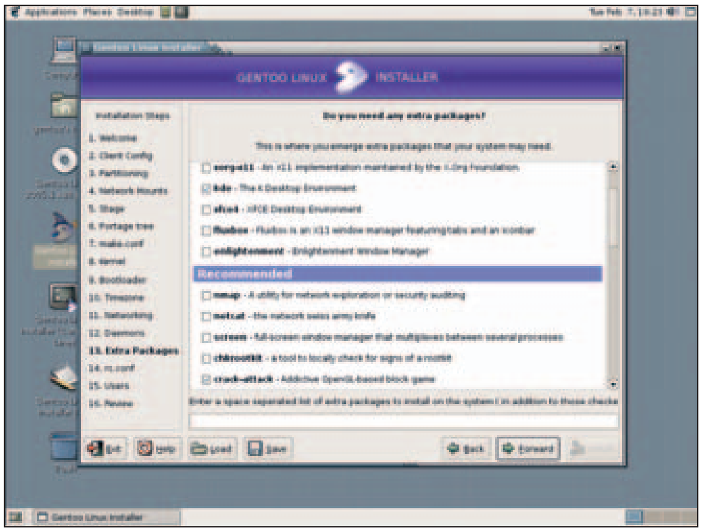
Gentoo не похож на большинство дистрибутивов Linux. Вместо готового набора приложений он предоставляет абсолютно все компоненты, из которых вы можете построить собственный дистрибутив и собрать его из исходных текстов, даже без команды на компиляцию. В связи с этим Gentoo не имеет версий. После установки он обновляется при появлении новых пакетов, поэтому любой пользователь Gentoo, читающий данный обзор, уже имеет систему новее, нежели рассматриваемая.

Установка Gentoo стала легче и быстрее, чем раньше – к огорчению элиты пользователей Gentoo. Теперь метод установки по умолчанию – это Stage 3, наличие CD с уже скомпилированными пакетами уменьшает время установки от дней до одного-двух часов. Кое-что еще приходится конфигурировать вручную, зато компьютеру уже не надо часами пыхтеть над компиляцией исходных текстов. В Stage 3 почти вся базовая система берется из tar-архивов, которые просто распаковываются на диск. Разбиение диска и компиляция ядра оставлены пользователю, так что дистрибутив все еще рукотворный, но блоки его укрупнились и их стало меньше. Установка Gentoo – прекрасное средство обучения, в основном благодаря несравненной документации и поддержке. При самостоятельной настройке системы необходима помощь со стороны, и от Gentoo вы ее получаете – это одно из главных достоинств проекта.

Новый графический инсталлятор, доступный в бета-версии, еще больше облегчает жизнь. Не ждите незатейливого «выбери-и-щелкни», как в SUSE, но зато некоторые вещи заметно ускорены, и можно, например, сохранить настройки установки и, установив систему на одной машине, без проблем перенести ее на другую. Есть также несколько графических утилит управления Gentoo, например, Porthole (менеджер пакетов на основе *GTK*) и набор модулей Центра Управления KDE.

Все под контролем

Управляемость – одна из основных составляющих Gentoo: при установке вы сами определяете опции и даже ключи оптимизации компилятора. Использование готовых бинарных пакетов убирает эту возможность, но только на этапе установки, чтобы вы могли быстро получить полностью рабочую среду. Полный контроль над уже готовой системой остается за вами, и даже больше чем раньше, – при желании систему можно минимизировать или максимизировать. После установки вы можете соединиться с сервером Gentoo, получить информацию о последних версиях, отредактировать настройки и всего одной командой перекомпилировать систему по вашим спецификациям. Времени потребуется немало, но процесс идет в фоновом режиме, так что ваша работа не прервется.



Графический инсталлятор Gentoo стал пригодным к использованию.

Portage (система управления пакетами в Gentoo) позаботится о загрузке исходных текстов, обеспечении зависимостей, конфигурации и компиляции – это ключ, предоставляющий пользователю контроль над ОС. Охватив почти 11000 пакетов, включающих поддержку многих коммерческих программ, *Portage* данного релиза значительно улучшился. Действительно, со времени прошлогоднего обзора Gentoo разработчики в основном занимались им, и в результате система стала более гибкой и простой в работе. Например, предусмотрен, как в Debian, выбор между стабильным и тестовым вариантом системы, причем *Portage* теперь умеет запускать их смесь, для которой можно указать основные пакеты стабильных версий плюс несколько находящихся в стадии тестирования.

Поскольку дистрибутив основан на исходных текстах, пакеты для разных архитектур появляются практически одновременно. Мы запускали Gentoo на системах i586, Athlon 64 и PPC G4; он доступен также для PPC 64 (G5), SPARC, Alpha, HP-PA и MIPS. Есть и неофициальные порты на другие системы, например, Xbox.

Если вы готовы потратить на начальную настройку чуть больше времени, чем обычно, дистрибутив Gentoo Linux, возможно, то, что вы ищете. Наиболее полный из дистрибутивов на основе исходных текстов, он к тому же прост в использовании и администрировании. Конечный результат – система Linux, выполняющая то, что вам нужно, так, как вы хотите, вне зависимости от решений других людей. LXF

Учебники Gentoo, форумы и списки рассылки предоставят вам всю необходимую информацию, и более того.

СОСТАВ GENTOO

На момент написания статьи...

- Ядро 2.6.15.3
- Gnome 2.12.0.1
- KDE 3.5.1
- *Firefox 1.5.0.1*
- *GCC 3.4.5**
- *OpenOffice.org 2.0.1*
- *X.org 6.8.2***

*GCC 4.0 доступен, но не используется по умолчанию.

ВЕРДИКТ LINUX FORMAT

ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	9/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	7/10
ДОКУМЕНТАЦИЯ	10/10

Если вы хотите принимать решения самостоятельно и не боитесь лишней работы, возможно, это ваш дистрибутив..

РЕЙТИНГ 9/10

СЕТЕВОЕ РЕЗЕРВИРОВАНИЕ

Arkeia Smart Backup 5.4.9

Это простое и гибкое приложение обеспечивает максимальную безболезненность процесса резервирования. Так что больше никаких оправданий, пишет **Грэм Моррисон**, лучше скажем хором: «Arkeia, мы сделаем это наилучшим образом».

САМОЕ ГЛАВНОЕ

Планировщик резервирования данных с сетевых машин. Альтернативы: *BakBone NetVault* или ваш собственный скрипт.

- **РАЗРАБОТЧИК:** Arkeia
- **САЙТ:** www.arkeia.com
- **ЦЕНА:** Первые 50Gb бесплатно. $100 за каждые 100Gb. Свыше 1Tb – $500 за каждые следующих 500Gb

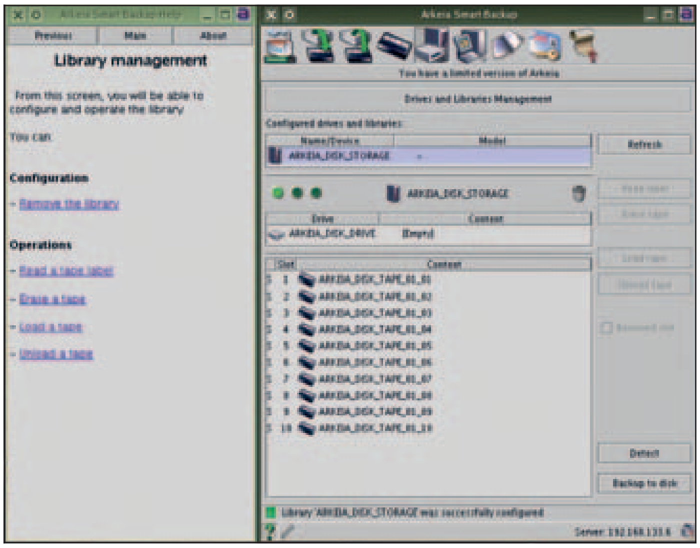
Как певец, переросший свою группу, марка ПО для резервирования *Arkeia* к 2002-му году затмила своего создателя, заставив фирму Knox Software сменить название на Arkeia Corp. Linux-решение для резервирования от Arkeia популярно и сегодня, но *Smart Backup* – новый подход, выбранный фирмой для предоставления малому и среднему бизнесу варианта резервирования с дифференцированной схемой оплаты. Цена программы пропорциональна объему информации, который необходимо защитить, и если он меньше 50 ГБ, то программу можно использовать бесплатно. Звучит неплохо, но из вздорности заметим, что эта схема не поможет вам сэкономить при росте сети.

Для решения, нацеленного на малый бизнес, количество имеющихся готовых пакетов ободряюще велико: охвачено всё от Slackware до SUSE, не считая пары 64-х битных систем. Для установки необходимо два пакета: сервер и графический клиент. В большинстве систем конфигурирование не требуется. Сервер стартует автоматически и может останавливаться и запускаться при помощи простого скрипта.

Клиент является Java-приложением и потому может запускаться на множестве платформ помимо Linux – это удобно для администраторов смешанных сетей и обеспечивает гибкий доступ к серверу. При первом запуске клиента необходимо ввести адрес сервера, после чего клиент соединяется с машиной-сервером и открывает главное окно.

Что же есть в этом квадратном окне? К счастью для новых пользователей, интерфейс хоть и серенький, но дружелюбный. Почти всю левую половину экрана занимает постоянно обновляемая панель справки, которая описывает опции, выбираемые в главном окне справа.

После задания своего пароля вам необходимо тут же сконфигурировать клиента. Тут можно запутаться, так как «клиент» в данном случае означает файловую систему, где сервер должен брать данные для резервирования, а не Java-приложение, используемое вами для управления сервером – больше похоже на удаленный графический пульт управления.



Благодаря модели клиент-сервер вы можете разворачивать и конфигурировать резервную копию из любого места, доступного через вашу сеть.

Копия ленты на ленту

В простейшем случае клиент для резервирования может быть той же машиной, что и сервер, например, при резервировании локальных файлов данных обычного почтового или файл-сервера.

Но истинная сила *Smart Backup* в доступе к удаленным файловым системам. При этом вам необходимо добавить клиента как удаленную машину. Определение клиента может быть делом нелегким, так как нужно указать полное имя хоста, но небольшое изменение файла **/etc/hosts** на обеих машинах может позволить в малых сетях работать без DNS. Использование только IP-адреса невозможно.

Smart Backup подразумевает подключение к серверу массива ленточных устройств резервирования. Если фирма их не использует (что имеет место в большинстве мелких фирм), можно также создать виртуальный массив в локальной файловой системе.

Запуск задания заключается в выборе клиента и перемещения по его файловой системе в поиске элементов для резервирования. Это может быть, например, база данных *MySQL* или раздел **/home**. Резервирование может выполняться вручную или автоматически по расписанию; предусмотрены накопительное (incremental) и разностное (differential) резервирование. Разностное копирует все файлы, изменившиеся с момента последнего полного резервирования; накопительное резервирование копирует только файлы, изменившиеся после предыдущего накопительного резервирования.

Другая приятная особенность *Smart Backup* – способность параллельного резервирования нескольких клиентов, облегчающая резервирование нескольких домашних папок, разбросанных по всей сети.

Процесс восстановления данных после сбоя также хорошо продуман. Просто выберите резервный файл, и *Smart Backup* уже будет точно знать, где он размещается. Затем выберите: восстановить все или просмотреть отдельные файлы и папки (можно восстановить их в другом месте). Удобная функция поиска поможет вам отловить все необычное.

Мы обнаружили, что *Smart Backup* весьма эффективен: резервирование раздутой группы домашних папок размером 100 МБ заняло менее 5 минут без значительной загрузки процессора. Приложение хорошо разработано и просто в использовании, что важно для создания грамотного режима резервирования. Это идеальное решение для малого бизнеса, использующего в сети Linux-машины. LXF

ВЕРДИКТ LINUX FORMAT

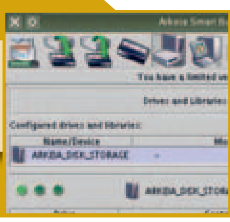
ФУНКЦИОНАЛЬНОСТЬ	8/10
ПРОИЗВОДИТЕЛЬНОСТЬ	8/10
ПРОСТОТА ИСПОЛЬЗОВАНИЯ	7/10
ЦЕНА/КАЧЕСТВО	7/10

Прекрасная утилита резервирования, идеальна для малого бизнеса, хотя схема оплаты не слишком-то экономична.

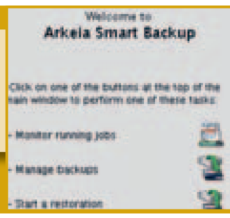
РЕЙТИНГ 8/10

ОДНИМ ВЗГЛЯДОМ

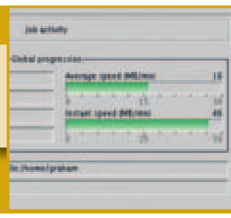
Все функции снабжены наглядными иконками.



Контекстная справка проводит вас по каждой опции.



Параллельное сетевое резервирование.



МОДЕЛИРОВАНИЕ УПРАВЛЕНИЯ АВИАКОМПАНИЕЙ

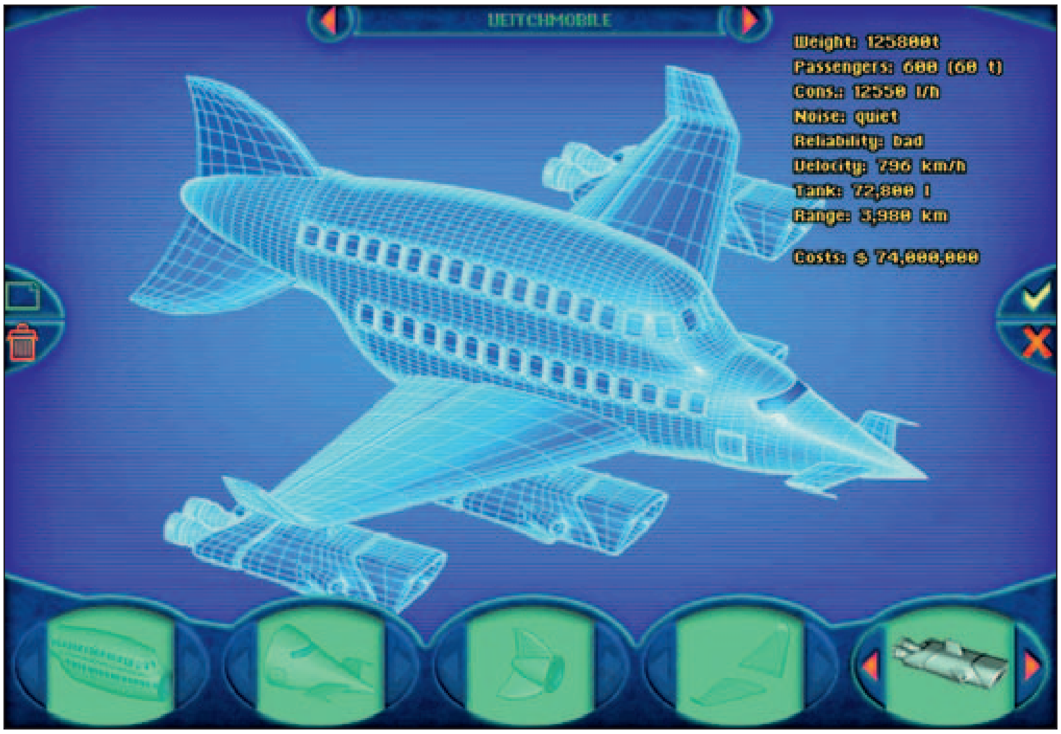
Airline Tycoon Deluxe

Тошно сидеть на приставном месте? Реформа самолетного кресла по **Полу Хадсону**: создание своей авиакомпании...

САМОЕ ГЛАВНОЕ

Экономическая игровая модель мира авиалиний. См. также: Software Tycoon, OpenTTD или FreeRails.

- **РАЗРАБОТЧИК:** RuneSoft
- **САЙТ:** www.tuxgames.com
- **ЦЕНА:** $40



Вейчмобиль: раздувшийся, вредный для глаз, гадски медленный – якобы вариант аэробуса.

Целыми днями упорные журналисты LXF снуют по миру умников, разыскивая новые проекты и людей, чье время пришло. В погоне за лучшими приложениями с открытым кодом мы летаем вокруг всего земного шара. Но с целью снижения себестоимости наш издатель настаивает на путешествиях эконом-классом (известным как зона острого тромбофлебита). Несколько часов перелета из города в город посвящаются злобному обдумыванию плана мести соседу спереди, который всего-то опустил спинку своего кресла, заодно расплющив ваши колени. Будь мы хозяевами авиакомпании, у всех были бы просторные места, обед из настоящей, а не синтетической курицы и Wi-Fi доступ.

После стольких лет подавления эмоций у нас появился шанс излить их, под маской *Airport Tycoon Deluxe* (ATD). Само слово 'tycoon' (магнат) в названии игры сразу же вызывает специфические чувства у игроков. Некоторые вспоминают славные деньки *Railroad Tycoon* и *Transport Tycoon* и хотят, чтобы их разработчик Microprose все еще был рядом. Другие могут припомнить только бесконечный поток тематических выпусков от эпигонов Microprose – *Zoo Tycoon, Mall Tycoon, Vegas Tycoon* и прочие. ATD недостает веселой оригинальности и искры творческого гения, как у *Transport Tycoon*, но все-таки она возвышается над остальной массой.

Формула Tycoon

Обычная Tycoon-игра следует стандартной формуле *Theme Park*. Во-первых, люди в ней изображаются как в мультиках, на базе псевдо-3D-графики. Во-вторых, предусматривается множество мелких штучек, по щелчку на которых происходит нечто смешное, вроде прихлопывания мухи или забредания не в тот туалет, с последующим визгом. И наконец, должна быть пара-тройка советчиков, на протяжении игры периодически выскакивающих с подсказками. В *ATD* все вышеперечисленное имеется, так что поначалу игра выглядит вполне типично. Но есть, есть у нее тузы в рукаве.

Ваша задача – управлять авиакомпанией в аэропорту, где работают еще три другие, конкурирующие компании. Вы можете покупать и продавать акции других компаний, но дело жизни для вас – ваша собственная. Если сумеете перевозить пассажиров и грузы между аэропортами всего мира, то процветание вам обеспечено, и большую часть времени вы будете проводить, составляя расписание полетов для вашего флота.

В игре приняты некоторые меры по снижению сложности, среди которых наиболее полезная возможность – «автопилот», он автоматически отгонит ваш самолет к следующей точке маршрута. Если вы уложились в срок, вы получаете жирный куш, не смогли – сумма штрафа намного превысит доход.

Для разнообразия, можно нанимать и увольнять пилотов, проводить рекламные кампании и обновлять парк самолетов. Обставлено забавно: вначале вы вынуждены покупать летающие гробы б/у в музее авиации, но вскоре замахнетесь и на прекрасные лайнеры последних моделей прямо с завода. Продвинутые игроки смогут даже разрабатывать собственные воздушные суда, укомплектовывая их креслами желаемого типа в индивидуальной обстановке.

Бизнес ведется в здании аэропорта, где течет довольно-таки скучная жизнь – спросите Мимино. Но вы быстро обнаруживаете целый подпольный мир подлянок, задуманных для подножки более успешным конкурентам.

Отчаянные меры

Самые безобидные из уловок – подсыпать соли в завтраки другой авиакомпании или подсунуть свою брошюру в зал ожидания конкурента, но более отчаянные игроки могут от злости и бомбу заложить. Выбор персонажей разработчиками вызывает недоумение. Вы покупаете и продаете керо-

Окно планирования полетов – ваша жизнь, запомните в нем каждый самолет в каждой стране. Вы его полюбите!

Каждое утро менеджер аэропорта проводит летучку и сообщает о положении вашей авиакомпании на фоне конкурентов.

Купите керосин у этого арабского джентльмена, налево кругом – и вперед, за планирование терактов. Увы, мы не шутим.

син у явно жуликоватого араба в аэропорту. Если ему что-нибудь подарить, он проведет вас в укромное место, где вас встретит другой араб, вооруженный до зубов; этот работает штатным воздушным террористом. Мы не видим оправдания подобным расовым стереотипам, и можно только надеяться, что скорейшие заплатки сделают игру менее оскорбительной.

Кроме того, *ATD* тянут назад неудачные попытки создания живого голосового окружения. Большинство встречных говорят по-английски с сильным южногерманским акцентом (RuneSoft и Spellbound Entertainment, создатели *ATD*, расположены в Баден-Вюртемберге, чтобы вы знали), и в результате непредвиденно комичны. (Пример: 'dough' (деньги) произносится как 'doff' (раздевайтесь)). С персонажами, озвученными настоящими англичанами, дело обстоит еще хуже: откуда-то берутся загадочные длинные паузы между словами и вставки фраз, произнесенных другими актерами. Например, можно предположить, что в последний момент поменяли названия авиакомпаний, потому что нормальная речь некоторых персонажей сбивается на этих названиях на легкий немецкий акцент, а затем становится прежней. Понятно, что авиалинии – международный бизнес, но все же это есть некарашо.

Мы и раньше играли в немецкие стратегии, и одна вещь раздражала нас постоянно: уровень микроуправления. По каким-то причинам в играх немецких фирм требуется внимание к каждой части вашего мира, причем мелочность – залог успеха. Например, когда мы рассматривали в *LXF57* игру *Northland*, аналогичную *Settlers*, то обнаружили, что в игре необходимо точное определение пола каждого младенца в вашей деревушке, в то время как вы пытаетесь отбиться от агрессоров-соседей.

Приятно, что в *ATD* такого не встретишь. Единственный минус этого рода – предоставляется столько цифр и графиков, что не понять, в чем ты неправ. В *Theme Park* можно было точно узнать, что думает каждый из ваших клиентов, и получить отзывы о ваших магазинах и экскурсиях. А в *ATD* практически нет контакта с людьми, летающими вашими рейсами, нет отчетов о качестве полетов – только минимальное количество советов по улучшению вашей сети. В результате при потере денег до вас не скоро доходит, из-за чего все приключилось; к решению вы придете только после длительного периода проб и ошибок.

Терминалы терминала: каждое место в игре находится в аэропорту, поэтому придется пошагать. Эх, было бы в Гатвике так же пусто…

Управление временем

В ATD хватает разнообразных миссий, но основной все-таки является свободная игра, не имеющая завершения. Вот где сложность становится поистине невыносимой: заниматься двумя самолетами в течение пяти дней в одном аэропорту уже непросто, а если арендовать трассы по всему миру и создавать огромный флот, придется заведовать 20 самолетами в 40 аэропортах. К сожалению, нельзя передать часть функций компьютеру, чтобы тот управлял полетами, а вы занимались стратегией, так что на определенном этапе компьютерный конкурент просто раздавит вас, вы уже не в силах будете с ним тягаться. Еще противнее, что вы покидаете «офис» в 18:00 и возвращаетесь в 9:00, так что рабочего времени элементарно не хватает на управление крупной фирмой.

Поскольку в *ATD* нет многопользовательского режима, то удовольствие получаешь только от построения успешной авиакомпании, да еще, быть может, надежды победить компьютер. Как ни странно, именно эта надежда час за часом удерживала нас за игрой. Графика не нова, музыка на среднем уровне, плоские шутки в стиле Tycoon едва вызывают улыбку – но в отсутствие альтернативы эта игра вполне способна заставить вас возвращаться к ней снова и снова, чтобы побить систему. Как сказал Эд Аккер (Ed Acker), бывший глава Air Florida: «Авиабизнес – это сущий наркотик». После нескольких часов игры вы будете дико ликовать, сумев на час ужать расписание.

ATD сама по себе не столь забавна, но каким-то образом наполняет вас стальным желанием биться до победы. Посмотрим правде в глаза: любая игра, способная всего за $40 завлечь вас на неограниченное время, полностью оправдывает себя [не будем забывать, что обзор делали в Великобритании, – прим. ред.]. Возможно, это не «deluxe» во всех смыслах этого слова, но она заставит вас жить и дышать самолетами. LXF

ВЕРДИКТ LINUX FORMAT

ИГРАБЕЛЬНОСТЬ	6/10
ГРАФИКА	4/10
ДОЛГОВЕЧНОСТЬ	8/10
ЦЕНА/КАЧЕСТВО	6/10

Выглядит как продукт автоматического Tycoon-генератора, но вы не раз к ней вернетесь.

РЕЙТИНГ 6/10

Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux.

СТАРЫЙ ДРУГ

Ладислав Боднар (Ladislav Bodnar) – основатель, хранитель, начальник, сотрудник DistroWatch.com.

Пару месяцев назад кое-кто из сообщества Linux выражал сомнение в достоинствах дистрибутива Slackware Linux. Приводя как аргумент удобство использования Mandriva, SUSE или Ubuntu, они утверждали, что немногие пользователи-новички захотят даже попробовать установить дистрибутив, где отсутствуют современные инструменты управления пакетами программ, а все настройки выполняются редактированием малопонятных текстовых файлов.

«SLACKWARE ВСЕГДА БУДЕТ ЛЮБИМЫМ ДИСТРИБУТИВОМ.»

Скептицизм вполне обоснован. Основанный Патриком Фолкердингом (Patrick Volkerding) в 1991-м году, Slackware всё ещё в значительной мере проект одного человека. Набор пакетов, распространяемый со Slackware, уменьшается, а web-сайт дистрибутива в течение года почти не обновлялся.

Несмотря на это, Slackware живет. Одни хвалят скорость этой небольшой и чисто сделанной системы, другие наслаждаются ее мощью, ибо дистрибутив Slackware больше всех остальных похож на Unix. Если что-то не ладится, в Slackware вы всегда знаете куда смотреть; а вот ошибку в дистрибутивах типа Mandriva или SUSE гораздо сложнее найти и исправить.

В известном смысле, Slackware напоминает текстовый редактор *Vi*. Он неинтуитивен и сложен, и многие предсказывали ему скорую кончину после появления современных графических интерфейсов. Но Vi посрамил этих пророков и по-прежнему процветает. Slackware вряд ли снова достигнет верхних строчек рейтинга (в своё время он захватил 90% рынка), но останется любимым дистрибутивом тех пользователей, кто понимает больше, чем мрачные прорицатели.

Fedora Core 5

Девять месяцев после последнего стабильного релиза – и прощай, шляпа!

Интервал между выпуском Fedora Core 5 и предыдущим релизом составил почти девять месяцев. Изначально предполагалось, что у этого дистрибутива будет полугодичный релиз-цикл, но недавняя реорганизация вкупе с ещё не завершившимся становлением Fedora Foundation удлинила разработку новой версии. Хотя дистрибутив немного запаздывает, его стоит подождать.

Что же нового в Fedora 5? Первым делом, видимо, бросится в глаза новая тема рабочего стола. У Fedora появился новый шикарный логотип, который красуется на обоях рабочего стола, установленных по умолчанию.

Fedora 5 включает версию Gnome 2.14 и новый инструмент обновления пакетов.

Большие изменения также произошли в инсталляторе *Anaconda*. Его интерфейс упрощён, а многие продвинутые опции упрятаны поглубже. Механизм диалога выбора пакетов *Anaconda* теперь привязан к *Yum* – менеджеру пакетов по умолчанию в Fedora, это позволяет более тонко управлять набором устанавливаемых приложений.

Другое интересное добавление к новому релизу – *Package Updater* (для краткости, Pup). Это несложный графический интерфейс к *Yum*, позволяющий пользователям устанавливать обновления безопасности и заплатки во время всей жизни дистрибутива. Дебютировали также несколько приложений Gnome, основанных на Mono (набор программ для разработки .NET-приложений). Сюда вошли *Beagle* (утилита поиска для рабочего стола), F-Spot (менеджер фотографий) и *Tomboy* (утилита ведения небольших заметок).

Не так заметны, но, возможно, более важны некоторые улучшения в новых версиях библиотеки GNU C (версия 2.4) и компиляторе *GCC* (версия 4.1). Новая версия *Glibc* – первое основное обновление библиотеки за три года. Библиотека существенно улучшена, а множество второстепенного кода (например, устаревшие LinuxThreads и поддержка необычных процессорных архитектур) переведено в разряд неподдерживаемых дополнительных расширений.

GCC 4.1 предлагает приложениям гораздо более надёжные механизмы защиты от переполнения буфера, применив идею «канареек» (canary words), позаимствованную у шахтёров, которые использовали птичек для определения недостатка кислорода в шахте. Если произошло переполнение буфера, то некоторая величина (канарейка) будет отличаться от стандартной.

Fedora Core 5 дорос до состояния крупного обновления, содержащего большое количество свежих программ, несколько новых инструментов и приложений, а также важные улучшения безопасности.

f-spot

REPRODUCED COURTESY: JOE SHAW

Beagle

Приложения Mono – прочная основа последних релизов дистрибутива.

Укрощение Slackware

VectorLinux 5.1 – Slackware, приученный к дому и офису

Сам по себе, Slackware годится не всем. VectorLinux, запущенный в 2000-м году, представляет собой производный от Slackware дистрибутив, дружелюбный к пользователю, который поддерживает совместимость с пакетами своего родителя.

В VectorLinux отличная документация и утилиты администрирования.

Основной целью проекта было создание операционной системы, способной работать на компьютерах с низкой производительностью, но разработчикам также удалось создать полноценную редакцию системы для офисного использования.

VectorLinux 5.1 SOHO (Small Office/Home Office – малый офис / домашний офис) был выпущен в январе. VectorLinux включает полноценный текстовый инсталлятор, помогающий пользователям конфигурировать систему во время установки, и великолепный автодетектор и автоконфигуратор аппаратного обеспечения. Сразу после установки SOHO 5.1 загружается прямо в рабочий стол KDE 3.4.2.

Возможно, наиболее ценный компонент VectorLinux – его графический инструмент администрирования по имени *Vasm*, или *Vector Administrative & Services Menu*. Эта простая утилита помогает пользователям менять пароли, устанавливать оконный менеджер по умолчанию (кроме KDE, в наличии есть *Xfce 4*) и настраивать Dmix для одновременного проигрывания нескольких аудиопотоков. В режиме суперпользователя появляются дополнительные опции, включая настройку сервисов и оборудования.

Другая замечательная программа называется *Gslapt* – графический интерфейс к утилите Slackware *slapt-get*. Эта продвинутая утилита управления пакетами по умолчанию настроена на соединение с репозитарием пакетов VectorLinux (который содержит солидный набор дополнительного ПО, не попавшего в инсталляционный диск). *Gslapt* исключительно удобна для установки и удаления пакетов, что придаёт дополнительную ценность дистрибутиву.

Опрятный и быстрый дистрибутив VectorLinux 5.1 SOHO доступен для бесплатного скачивания или на официальном компакт-диске в онлайновом магазине проекта.

www.vectorlinux.com

ХИТ-ПАРАД ДИСТРИБУТИВОВ

10 самых популярных страниц на DistroWatch.com в январе (среднее число визитов в день)

1	**Ubuntu Linux**	2,616	⇔
2	**SUSE Linux**	1,702	⇔
3	**Fedora Core**	1,291	↑
4	**Mandriva Linux**	1,199	↓
5	**Mepis Linux**	1,125	⇔
6	**Damn Small Linux**	989	⇔
7	**Debian GNU/Linux**	772	⇔
8	**Gentoo**	629	↑
9	**VectorLinux**	603	↑
10	**Knoppix**	602	↓

DistroWatch.com отслеживает популярность дистрибутивов, основываясь на количестве посещений соответствующих им страниц сайта. Хотя эти цифры и не отражают настоящее количество инсталляций, они являются индикатором популярности дистрибутива на данный момент времени.

На страже сервера

EnGarde Secure Linux 3 позволяет управлять сервером через web-браузер

Ищете безопасный серверный вариант дистрибутива? Вы не ошибётесь, если выберете EnGarde Secure Linux. Это продукт Guardian Digital, находящийся в разработке с 2001-го года. Версия 3 свободной редакции дистрибутива была выпущена в конце прошлого года, а новые обновления появляются с регулярными интервалами. Последняя версия EnGarde 3.0.3 доступна как для процессоров i586, так и для x86_64.

Центральное место в EnGarde занимает впечатляющее приложение *WebTool* – всеобъемлющий web-инструмент для администрирования. WebTool написан на Perl, что позволяет ему с лёгкостью настраивать большинство аспектов функционирования сервера, включая web, почту, FTP, DNS, брандмауэр, SSH и другие сервисы. С его помощью можно настроить дату и время, пользователей и даже UPS (источник бесперебойного питания).

WebTool позволяет настраивать параметры безопасности: систему обнаружения вторжений, сканирования сети или функциональность SELinux. Системные журналы также можно просматривать из WebTool. Обновления безопасности и исправления ошибок становятся доступны после получения бесплатного активационного кода с сайта проекта.

Новая возможность в EnGarde 3 – режим Live CD, в нем пользователи могут попробовать продукт перед полноценной установкой на жёсткий диск. Просто загрузитесь с компакт-диска и следуйте инструкциям по установке паролей и настройке сети, затем подсоединяйтесь к серверу из любого браузера. EnGarde 3 включает обычное для серверов ПО – Apache 2, PHP 4, *MySQL 5.0*, *Bind 9*, *Postfix* и SSH. Фирма не только предоставляет отличный дистрибутив, но и поддерживает полезный web-сайт, посвящённый проблемам безопасности **www.linuxsecurity.com**.

***WebTool* позволяет выбрать модули администрирования из выпадающего меню.**

Последний релиз EnGarde Secure Linux, без сомнения, один из самых дружественных пользователю серверных дистрибутивов на данный момент.

www.engardelinux.org. LXF

КАКОЙ ДИСТРИБУТИВ БОЛЬШЕ ВСЕХ?

Почему Debian предлагает в 30 раз больше пакетов, чем Slackware

Обладая почти двадцатью тысячами пакетов, Debian GNU/Linux несомненно самый большой дистрибутив Linux – сравните его со Slackware, который предлагает всего около шестисот пакетов, причём их число уменьшается после каждого стабильного релиза. Однако удивляться тут нечему, поскольку поддержкой Debian занимается почти тысяча активных разработчиков, тогда как Slackware по сути представляет собой дистрибутив одного человека.

Цифры говорят сами за себя, однако сравнение не совсем честное. Некоторые дистрибутивы, особенно Debian и Mandriva, разделяют свои бинарные пакеты на несколько частей, помещая библиотеки и заголовочные файлы в различные пакеты.

Дистрибутив	Пакеты	Комментарий
Debian GNU/Linux	19,386	все пакеты в /sid
FreeBSD	12,323	все порты в каталоге /current
Mandriva Linux	12,135	/main: 4,481, /contrib: 7,654
Gentoo	10,776	все пакеты в /portage
SUSE Linux	6,343	/factory: 6,271, /extra: 72
Fedora Core	5,368	/core: 2,226, /extras: 3,142
Slackware	613	/current: 473, /extra: 140

Сравнение

Каждый месяц мы просматриваем тонны ПО, чтобы вам не приходилось заниматься этим самостоятельно

DVD-ПЛЕЙЕРЫ

НА ДИСКЕ

Грэм Моррисон разжился самым длинным сериалом «Друзья», чтобы испытать лучшие плейеры DVD для Linux.

Пару лет назад все программное обеспечение для просмотра DVD под Linux сводилось к смехотворным потугам. Отсутствовал даже слабый намек на конкуренцию с программами коммерческих ОС. Причина в том, что DVD представляет собой смесь нескольких важных технологий, ни одну из которых нельзя было без проблем реализовать средствами свободного ПО. К счастью, обстоятельства изменились: сегодня выбор плейеров намного богаче, и все они работают хорошо.

Хотя и с двойным дном. Почти каждый DVD-плейер, который мы присмотрели для обзора, является скорее оболочкой, чем полновесным приложением: большинство из них, вместо тяжкого труда по воспроизведению видео с диска, паразитирует на одной из двух утилит: *Xine* и *MPlayer*, и, как недавно подметили на форуме *LXF*, «вся разница в пользовательском интерфейсе».

Это изречение стало нашим руководством к действию. Хотя все представленные здесь приложения не бедны умными функциями, мы судили их по одному критерию – насколько удобно с их помощью смотреть DVD. Важным критерием стал дизайн: при просмотре существенную роль играет количество включенных функций или расположение DVD-меню.

Даже простейшее воспроизведение диска потребовало от разработчиков геркулесовых усилий. Чтобы программа стала похожа на DVD-плейер, прикорнувший под вашим телевизором, нужно обеспечить пару-тройку обязательных свойств. Крайне важная совместимость с собственным интерактивным меню DVD вошла в строй только в течение нескольких последних лет. В большинстве современных DVD присутствуют и другие продвинутые функции, например, прослушивание разных аудиотреков или выбор угла зрения, и эти функции должны отражаться в соответствующем ПО.

На качество просмотра DVD влияют и такие факторы, как хорошая ответная реакция интерфейса и легкость возврата к выбранному моменту фильма. В результате одно приложение может сильно отличаться от другого. А победителем будет признано приложение, которое позволит вам выжать максимум из просмотра вашей коллекции DVD, затратив минимум усилий.

УЧАСТНИКИ СРАВНЕНИЯ

GXine ... 24
Kaffeine ... 22
KMPlayer ... 24
MPlayer ... 22
Totem ... 24
VLC ... 23
Xine ... 21

Xine

Произносится, по-видимому, как «гезин».

• **ВЕРСИЯ:** 0.99.4 • **WEB:** http://xinehq.de • **Цена:** Бесплатно по лицензии GPL

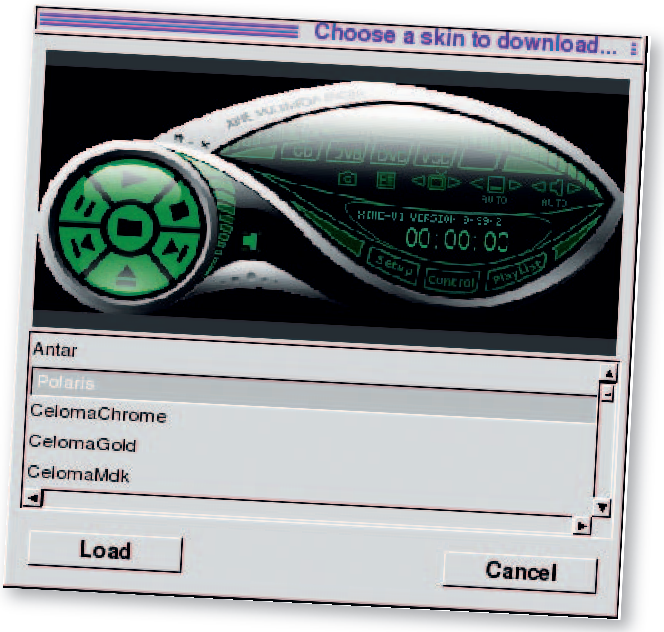
Большинство пользователей Linux так или иначе сталкиваются с *Xine*. Именно это приложение скрывается за большинством мультимедийных функций, предлагаемых типичным дистрибутивом Linux, так что избежать встречи с ним практически невозможно. *Xine* больше других повезло с командной строкой, и создан он так, что большинство других приложений легко к нему подключить. Результат – большинство DVD-плейеров в Linux не имеют собственного движка, а просто обращаются к *Xine*.

Xine интенсивно использует библиотеки (чего и следует ожидать от столь широко эксплуатируемого приложения). Причем не только те, которые относятся к воспроизведению DVD, а еще и отвечающие за другие форматы: *Xine* от потокового аудио до проприетарных кодеков Windows. Все, что вам нужно – это… библиотека (All you need is library – намек на известную песню All you need is love. Прим.перев.), мигом смекнете вы, попытавшись запустить *Xine* без пользовательского интерфейса и увидев только несметное количество страниц опций для выполнения мыслимых и немыслимых требований, например, конвертирования видео на лету или добавления каких-либо эффектов при воспроизведении. Для определения местонахождения *Xine* использует общий Локатор ресурсов медиа (Media Resource Locator), или MRL. Большинство других плейеров используют тот же формат, он просто дает вам возможность включить нужное устройство в имя файла. Чтобы посмотреть DVD, вы набираете **Xine dvd://** – и перед вами появляется окно воспроизведения с фильмом.

Желаете воспользоваться графическим интерфейсом пользователя (GUI)? Это не сложнее нажатия средней кнопки мыши. По умолчанию установлена панель, сверкающая якобы хромированными деталями, но она может «сменить шкуру» при помощи хитроумной системы загрузки из Интернета: просто кликните правой кнопкой мыши по окну воспроизведения и выберите в меню установок Просмотрщик (Skin Browser). В основном торжествует стиль конца 90-х – попытка художников воспроизвести солидный дизайн настоящих японских DVD-плейеров; не самое хорошее употребление экранного пространства, но лучше, чем вообще ничего. Странным образом, такой дизайн способствует скромному обаянию *Xine*.

Xine всех времен

Воспроизведение и производительность приложения отличные. Абсолютно все DVD-меню заработали в *Xine* (как и во всех плейерах, основанных на *Xine*), а количество опций конфигурации просто огромно.



Если вам не хватает интерфейсов в стиле 90-х, загрузите их – таких в *Xine* не один десяток.

Пользователей-новичков оно, конечно, может сбить с толку, но конфигурация GUI ограничивает управление интерфейсом в зависимости от вашей продвинутости. Выберите настройку «Новичок» – и вы сможете добавлять простые эффекты или изменять громкость звука. Настройка на Бывалого выдаст почти столько же опций, сколько предусмотрено для компиляции ядра Linux. Возьмите, к примеру, звук. Используя *Xine*, можно смикшировать многоканальный звуковой фон для любого количества динамиков (даже для наушников), а можно направить цифровой аудиопоток прямо на ваши устройства аудио. Это единственный практический способ поддержки Dolby surround под Linux, поскольку и сама природа открытого исходного кода, и затраты не позволяют использовать соответствующие программы.

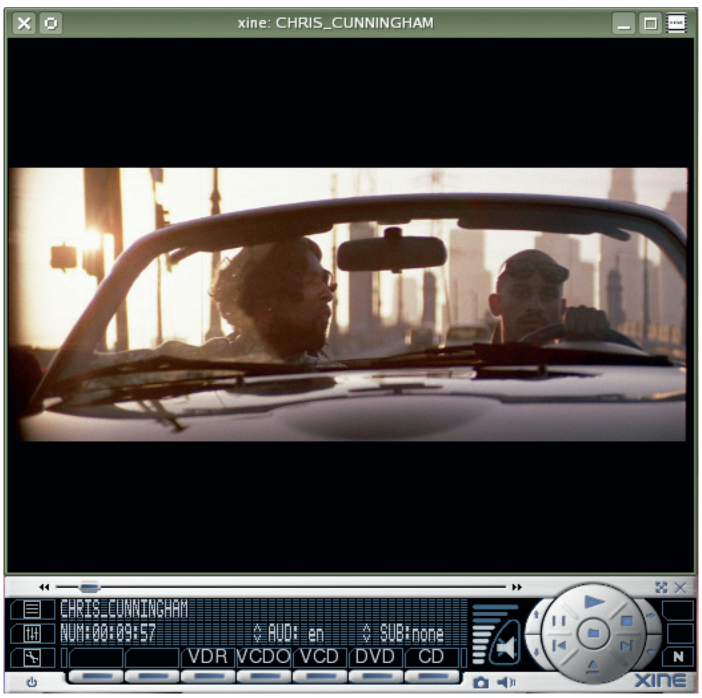
Изобилием опций отличаются и другие аспекты приложения, и этот подход – «чем больше, тем лучше» – пропадает впустую, когда требуется просто каждодневный просмотр киношек. Опций в *Xine* ну очень много, есть даже такие, которые относятся к несуществующим вещам: например, меню эпизодов DVD рассчитано на количество эпизодов от 1 до 30, хотя на диске помещается только 13. Как легко можно представить, это иногда превращает поиск нужного эпизода или подменю в пальбу наугад.

Поскольку *Xine* является сердцевиной многих других DVD-плейеров, очень важно, что здесь правильно воспринимается функциональность DVD. Это означает – и этим не может похвастаться *MPlayer* – хорошую поддержку главного меню DVD, в котором выбирается язык воспроизведения или эпизод, с которого следует начать, когда диск вставлен в дисковод. Раньше это всегда было проблемой, но команда разработчиков *Xine* умудрилась за последние 18 месяцев ликвидировать прорыв. В наших тестах мы использовали самые разные DVD, но *Xine* выдержал проверку, ни разу не споткнувшись. То же самое можно сказать и о других функциях DVD, таких, как поддержка субтитров, настройки альтернативного угла зрения или аудио потоков. Может быть, иногда функции трудно отыскать, но вся функциональность на месте, и это делает *Xine* превосходным движком DVD-плейера.

Забавные фильтры

Еще одно замечательное свойство *Xine* – использование фильтров для обработки аудио- и видеопотоков в реальном времени. Есть несколько заумные эффекты, например, изменение тона аудио или размывание изображения, но некоторые можно действительно применить с пользой, например, продвинутые методы борьбы с «гребенкой» (anti-interlace routines). С помощью основного интерфейса приложения легко варьировать яркость, контраст и насыщенность видео, но для аудио потока ничего подобного вы не найдете. Об этом особенно сожалеешь, когда смотришь что-то в общественном транспорте.

Жаль, что мы не можем дать *Xine* более высокую оценку: ведь программа «пашет» почти на все приложения из нашего Сравнения. Только благодаря команде разработчиков *Xine* многие другие могут создать собственный превосходный пользовательский интерфейс, совершенно не задумываясь о проблемах с воспроизведением фильма.



Xine **прославился как движок, но отлично работает и сам по себе.**

ВЕРДИКТ LINUX FORMAT

Краеугольный камень воспроизведения DVD в Linux, но интерфейсы бывают и получше.

РЕЙТИНГ 7/10

MPlayer

Еще один плейер на все руки.

• **ВЕРСИЯ:** 1.0 • **WEB:** www.mplayerhq.hu
• **Цена:** Бесплатно по лицензии GPL

Несмотря на все опции *Mplayer*, от оболочки никуда не деться.

***MPlayer* – еще один тяжеловес** среди DVD-плейеров для Linux, и у него много общего с *Xine*. В основном, это инструмент командной строки, лопающийся от функций, и многие приложения покушаются на его функциональность, беря его в качестве своего движка.

Большое достоинство *MPlayer* – поддержка изрядного количества видеодрайверов, в ассортименте от функционального до эзотерического. Например, вы можете смотреть фильм прямо во framebuffer'е, не запуская сессию *X Window*. На практике это означает, что можно обойтись совершенно убогой машиной, которая не тянет даже приличный GUI – программа способна воспроизводить DVD на старушках с процессором 200–300 МГц.

Другие драйверы видео вывода включают в себя отличную ускоренную опцию вывода 'xv', использующую оборудование для компенсации движения, которое имеется на большинстве современных графических карт; один из драйверов пользуется графической библиотекой SDL – поэтому *MPlayer* так часто выбирается для устройств, имеющих поддержку SDL.

Не все драйверы так уж полезны: два драйвера вывода, в частности, абсолютно пустяковые. Драйвер Brightlights управляет матрицей освещения, и им пользовались для наведения узоров на стенах офисов; а ASCII Output – это провоцирующая дикую головную боль версия исходного потока видео, трансформированного в текст. На секунду – забавно, но не дольше. По крайней мере, если хотите поджарить мозги в *Xine*, вам потребуется внешняя библиотека.

Командная строка *MPlayer* очень похожа на *Xine*. Они оба используют медиалокатор для указания местоположения (например, **dvd://**), и здесь достаточно опций, чтобы удовлетворить любой вкус. *MPlayer* зарабатывает очки благодаря опции для выбора эпизодов DVD из командной строки. Однако, в отличие от *Xine*, он даже не пытается обеспечить простой пользовательский интерфейс для тех, кто в нем нуждается. (Вообще-то, это не так. В состав *MPlayer* входит графический проигрыватель g*MPlayer*, базирующийся на gtk-1.2. Чтобы собрать его, необходимо использовать опцию **--enable-gui** сценария **configure**. По умолчанию эта функция отключена, – прим.ред.) Чтобы получить его, вам придется обратиться к одному из множества приложений, взявших *MPlayer* в качестве движка. Когда вы начинаете работать в приложении, оно в основном делает то, чего вы от него ждете. Однако самую большую проблему мы приберегли до конца обзора. *MPlayer* не поддерживает меню DVD! Это полнейший нокаут – по сравнению с его конкурентом с труднопроизносимым имечком.

ВЕРДИКТ LINUX FORMAT

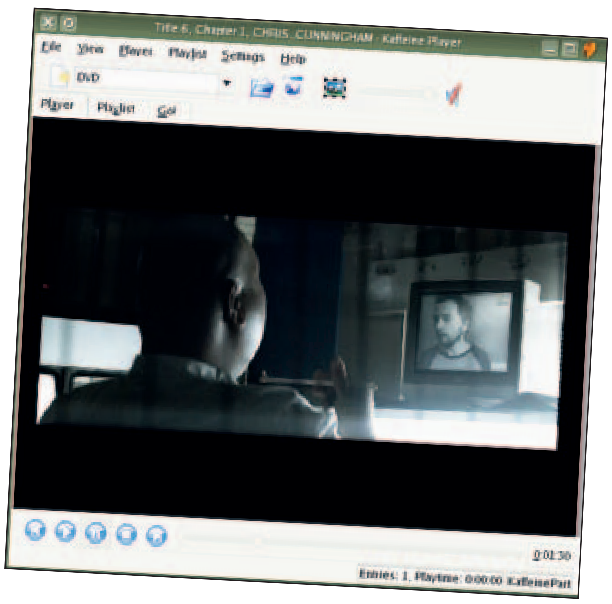
Функций больше, чем у конференц-центра, но для DVD – не лучший выбор.

РЕЙТИНГ 6/10

Kaffeine

Медиа-плейер KDE, о котором много шумели.

• **ВЕРСИЯ:** 0.7.1 • **WEB:** http://kaffeine.sourceforge.net
• **Цена:** Бесплатно по лицензии GPL



***Kaffeine* может работать и из окна браузера.**

Начав весьма скромно – в стиле «еще один медиа плейер», *Kaffeine* вырос до такой степени, что стал включаться по умолчанию во многие инсталляции KDE. Он может работать практически с любым форматом, какой ни предложи, и все это благодаря вездесущей библиотеке *Xine*. *Kaffeine* стал одним из новейших приложений KDE, которые добили наконец эту ужасную аудиомашину *Arts* и ввели поддержку воспроизведения аудио с использованием *GStreamer*.

Когда вы впервые запускаете *Kaffeine*, вам выдается полный обзор вашей системы, включая такую информацию, как DMA-режим вашего оптического привода и наличие библиотек, необходимых для воспроизведения DVD: и то, и другое может повлиять на воспроизведение. Главное окно *Kaffeine* не зависит от формата данных. Звучит здорово, но на деле выходит, что GUI *Kaffeine* от скуки на все руки, а толком ничего не умеет. Опции главного окна больше подходят для воспроизведения музыкальных CD; основные функции DVD приходится выискивать.

Отчасти причина тут в том, что *Kaffeine* заодно представляет собой попытку стать музыкальным плейером, онлайн-радиоприемником и альтернативой *RealPlayer* в одной посуде. Впрочем, имеется простое окошко «Go», где находятся иконки для всех основных задач, и в нем надо выбрать Open DVD, только и всего.

Благодаря библиотеке *Xine*, воспроизведение фильмов безупречно. Пользовательский интерфейс ненавязчив, но кнопки управления не выходят за рамки классических Воспроизведение/пауза (play/pause), Стоп (stop), Назад (previous) и Вперед (next) – не помешали бы Прокрутка вперед (FF) и Назад (REW). Все эти – и многие другие – функции доступны в меню по щелчку правой кнопкой. Предусмотрен простой доступ к аудио каналам DVD, субтитрам и содержанию, но как и в самом *Xine*, все наименования зашифрованы – и это серьезно усложняет поиск необходимой опции. Ну как догадаться, что Ch 2 означает субтитры?

Меню DVD можно привязать к клавише – хороший способ выбирать эпизод с любого места, а позиция в фильме относительно легко устанавливается ползунком. В общем, *Kaffeine* – отличный инструмент, куда лучше, чем использование *Xine* из командной строки, но пользовательский интерфейс пострадал от попытки объять необъятное.

ВЕРДИКТ LINUX FORMAT

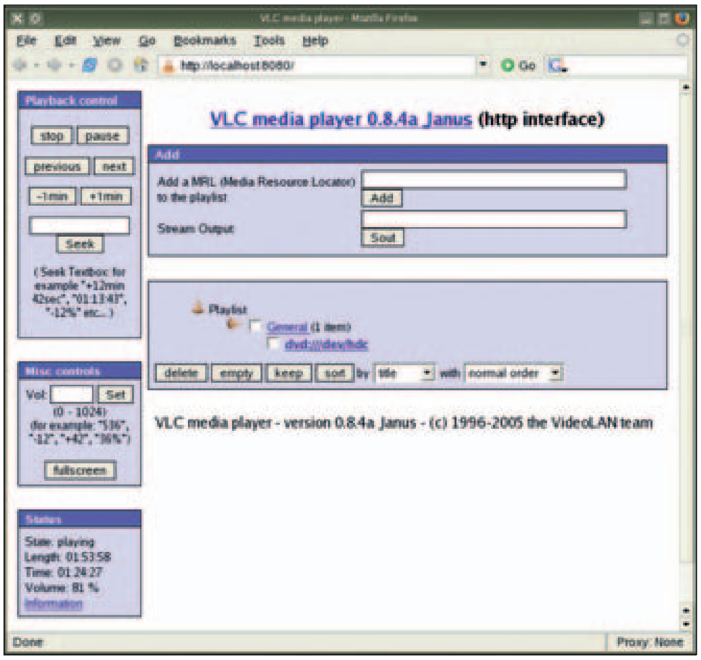
Один из лучших виденных нами плейеров, но DVD надо бы уделить больше внимания.

РЕЙТИНГ 7/10

VLC

Исключительный кросс-платформенный медиа-плейер.

• **ВЕРСИЯ:** 0.8.4 • **WEB:** www.videolan.org • **Цена:** Бесплатно по лицензии GPL



С виду *VLC* похож на другие приложения, но имеет свои козыри.

***VLC* – один из наиболее совместимых** медиа-плейеров, какие только бывают на свете. Он подходит не только для Linux, но и для Windows и OS X. Работает почти с любым типом медиа, даже с файлами плохого качества и поврежденными файлами, и зачастую *VLC* придает файлу смысл уже тем, что может его воспроизвести – все благодаря тому, что для своей работы *VLC* использует часть проекта *FFMPEG* вместо *MPlayer* или *Xine*. Один из положительных эффектов подобного выбора – *VLC* потребляет меньше ресурсов системы, чем его конкуренты; вдобавок работа с ним вносит приятное разнообразие.

VLC – еще одна утилита, чей духовный оплот – интерфейс командной строки. В ней почти столько же функций, сколько их в *Xine* или *MPlayer*, но при запуске приложения вы видите очень простое окно с полной навигацией и управлением. Запрос на открытие диска (Open Disc requester) из файлового меню использует несколько запутанную схему окон *VLC*, в которой все схожие функции (например, выбор медиа) сгруппированы на вкладках одного окна, и перемещение между опцией открытия файла и DVD означает смену вкладки – возможно, вы ожидали не совсем того. Однако *VLC* – единственное из рассмотренных нами приложений, которое способно распознать расположение вашего DVD-привода, если он не был обнаружен автоматически.

Меню DVD работает исключительно хорошо; нам не удалось заставить *VLC* совершить ошибку. Здесь есть даже опция, позволяющая пропустить меню и перейти прямо к показу – вещь полезная. Воспроизведение отличается наилучшей реакцией среди всех плейеров в нашем Сравнении, и просмотреть диск можно весьма быстро. Учитывая, что плейер не использует библиотек из *Xine* или *MPlayer*, это можно рассматривать как большое достижение. Выбор алгоритмов деинтерлейсинга (борьбы с «гребенкой») просто отличный, что весьма существенно для просмотра фильмов на жидкокристаллическом экране, а качество воспроизведения бросает вызов соперникам. Поддержка звука особенно хороша, поскольку имеется достойный выбор предварительных настроек эквалайзера, которые можно применять к потоку аудио в реальном времени.

Вывод изображения на экран тоже эффективен, а возможность перемещаться по списку воспроизведения с помощью нескольких простейших движений мышью является уникальной способностью *VLC*, которой не обладают другие приложения.

Трансляция видео

Странное название *VLC* всего-навсего сокращение от *VideoLan Client*. Изначально он должен был стать клиентской стороной утилиты, предназначенной для трансляции видео по локальной сети. Развитие сервера застопорилось, но упомянутая функция до сих пор очень сильно выражена в *VLC*; и, несмотря на наличие в названии слова «клиент» (client), *VLC* умеет и передавать, и принимать видеопотоки. Оба клиента должны работать на одной версии, и весь вопрос лишь в том, будет ли видео поток транслироваться на определенный интернет-адрес или на весь домен. Используется мультикаст UDP/RDP, совместимый с другими клиентами, например, *RealPlayer* или *Spook* – видеосервером, доступным для Linux. Вы можете запустить обслуживающий клиент на машине с DVD-плейером, и смотреть видео с другой машины, на которой плейера нет. Это забавно и действительно полезно.

> «VLC – ЕЩЕ ОДНА УТИЛИТА, ЧЕЙ ДУХОВНЫЙ ОПЛОТ – КОМАНДНАЯ СТРОКА.»

Оборотной стороной всей этой функциональности являются некоторые проблемы со стабильностью, особенно при переходе к разным эпизодам во время просмотра фильма. Запуск Расширенного интерфейса (Extended Interface) ухудшает ситуацию, а это, увы, означает, что отличное управление видео в реальном времени вовсе не так полезно, как могло бы быть. Остается, естественно, утешение, что стабильность – дело наживное, и мы от души надеемся, что команде разработчиков *VLC* удастся изгнать немногочисленные оставшиеся ошибки.

Когда вы смотрите DVD с использованием *VLC*, меню подгоняется под конкретный заголовок, и вам не придется плутать в безымянных пунктах меню, чтобы переключиться на другую звуковую дорожку. Все приложения, рассмотренные в этом Сравнении, от *Xine* до *MPlayer*, выводят информацию на экран во время просмотра фильма по-своему, но лучше всех это удается *VLC*: здесь вам не приходится все время возвращаться к основному окну, как в некоторых других приложениях. Кстати, о главном окне – *VLC* располагает одним из лучших интерфейсов с поддержкой «шкурок», которые можно скачать из Интернета. Некоторые – наиболее приятные и оригинальные – варианты оформления в кои-то веки основаны не на старомодном дизайне реальных DVD-плейеров.

В общем и целом, сочетание сетевого вещания, функциональности и дизайна интерфейса превращают *VLC* в соперника, которого трудно превзойти.

ВЕРДИКТ LINUX FORMAT

Лучшее воспроизведение DVD, с действительно полезной функцией потокового вещания.

РЕЙТИНГ 8/10

Интерфейс можно изменять в соответствии с самым изысканным вкусом

Totem

Наступление Gnome на движок Xine.

• **ВЕРСИЯ:** 1.3 • **WEB:** www.gnome.org/projects/totem
• **Цена:** Бесплатно по лицензии GPL

Подход простой и рабочий.

Totem* – эквивалент KDE *Kaffeine для Gnome. Это – официальный плейер Gnome для просмотра фильмов. Как и *Kaffeine*, он использует *Xine* для воспроизведения видео и *GStreamer* для аудио. Totem – часть приложений нового поколения в Gnome, в которых украшения опций конфигурации и пользовательского интерфейса сведены к минимуму. И этого нельзя не заметить уже с первого запуска программы. Вместо подхода, характерного для *Kaffeine* – «вот что вы можете сделать», *Totem* предоставляет вам только возможность посмотреть DVD – если в приводе есть диск. Нет, еще добавляет название диска в файловое меню.

В режиме просмотра наблюдается та же лаконичность дизайна: имеются основные навигационные опции, но при этом никаких раздутых пунктов меню (как правило, шествующих рука об руку с приложениями, которые используют библиотеку *Xine*), и большинство опций работает нормально. Единственный пример: выбор аудиоканалов небольшой, но достойный. С другой стороны, мы не смогли найти субтитры в собственном меню *Totem*, а в меню DVD они были. С другими плейерами такой проблемы не возникало.

ВЕРДИКТ LINUX FORMAT

Без него Gnome не был бы Gnome-ом.

РЕЙТИНГ 7/10

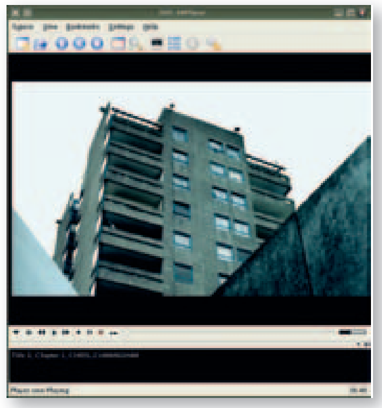
●●●●●●●○○○

KMPlayer

Попытка умерить фонтан опций MPlayer'а.

• **ВЕРСИЯ:** 0.9.2 • **WEB:** http://kmplayer.kde.org • **Цена:** Бесплатно по лицензии GPL

***KMPlayer* тоже поддерживает потоковое видео.**



Это приложение уже не самое новое, и изначально разрабатывалось с целью упростить навигацию по сотням опций *MPlayer* благодаря использованию несложного пользовательского интерфейса. За время, прошедшее с первого релиза, его дизайн не особо изменился; вообще-то он сильно напоминает *GXine*. Украшений крайне мало, а главное окно оснащено очень полезным информационным окошком под экраном воспроизведения видео.

В отличие от всех прочих приложений, включая *Xine*, панель конфигурации не прячет расширенных опций от новичка. Даже те, кто пользуется программой впервые, получают доступ ко всей гамме параметров, включая такие, как 'engine.performance.memcpy_method' или 'video.device.xv_pitch_alignment' – здорово, но непонятно. Например, режим интерлейсинга по умолчанию не очень хорошо работает на LCD-дисплее, и вам придется открывать ящик Пандоры в лице панели конфигурации, чтобы изменить метод отображения – во всех остальных приложениях эта опция доступна в простом меню. Навигация DVD тоже сыровата, и состоит из единственного пункта меню. *KMPlayer* использует библиотеку *Xine* [а также *MPlayer* или *Gstreamer*, на выбор, – прим.ред.], поэтому меню диска работает хорошо, но частенько возникают сложности с отслеживанием вашего местоположения на диске, в результате тормозится переход вперед и назад между эпизодами.

ВЕРДИКТ LINUX FORMAT

Хорош как движок, но не для кино.

РЕЙТИНГ 5/10

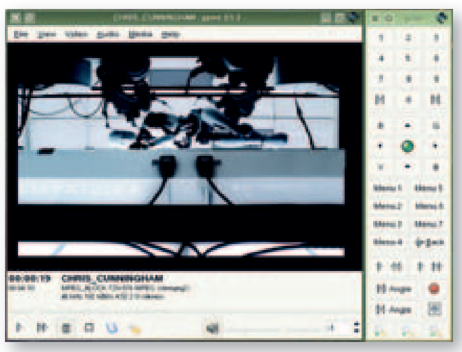
●●●●●○○○○○

GXine

А не станет ли это «еще одним интерфейсом с Xine»?

• **ВЕРСИЯ:** 0.5.3 • **WEB:** http://xinehq.de • **Цена:** Бесплатно по лицензии GPL

Вам не понадобится отдельный пульт дистанционного управления для *Gxine*.



Не трудно догадаться, что *GXine* – это медиа-плейер для Gnome, который использует библиотеку *Xine*. Несколько лет он находился в стадии медленной, но неуклонной разработки, а сейчас превратился в стабильное и понятное приложение. Дизайн интерфейса спартанский, но не до такой степени, как у *Totem*. Основное окно дополнено полезной панелью «деталей» внизу окна, которая сообщает вам разную ценную информацию, например, название DVD, размер видеопотока и качество аудио. *GXine* располагает большинством опций конфигурации *Xine*, позволяющих добавить фильтры пост-обработки к видео и аудио.

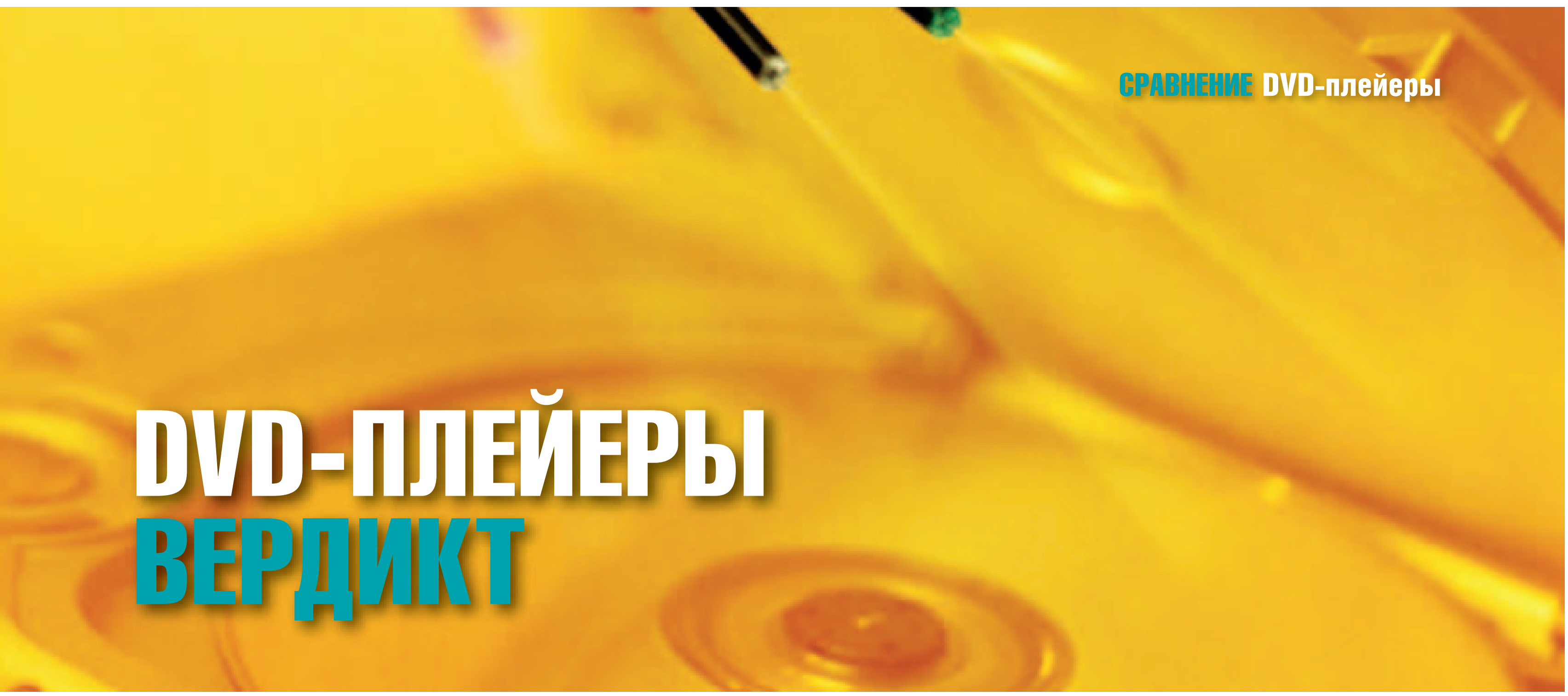
Основным отличием от прочих приложений для воспроизведения DVD является наличие виртуального дистанционного управления, которое выполнено в виде окна, где скопированы все основные кнопки, обычно имеющиеся на пульте дистанционного управления DVD-плейера: имеется маленький виртуальный direction pad, а также кнопки для уменьшения и увеличения скорости воспроизведения. Идея, может, и не кажется хорошей, но работает хорошо. Вам не нужно шарить по меню, разыскивая функции, которые – как вы точно знаете – обязаны там быть: вместо этого вы просто пользуетесь виртуальным пультом ДУ точно так же, как вы бы пользовались настоящим пультом у настоящего DVD-плейера. А еще это окно может оставаться открытым во время воспроизведения фильма в полноэкранном режиме, и это намного удобнее, чем обычный пользовательский интерфейс.

ВЕРДИКТ LINUX FORMAT

Одно из лучших приложений, просто превосходное, если вам нужно побольше деталей, чем в Totem.

РЕЙТИНГ 7/10

●●●●●●●○○○



DVD-ПЛЕЙЕРЫ ВЕРДИКТ

В процессе обзора этих достойных DVD-плейеров нам открылась интересная вещь: совершенно невозможно предсказать, каким окажется приложение. Как будто бы каждое приложение живет своей собственной изолированной жизнью на отдельном острове архипелага, объединенного технологией движка, и придерживается собственного подхода к пользовательскому интерфейсу. Если вам нужно легкое приложение, работа в котором не требует мыслительных усилий, то вам подойдет *Kaffeine* для KDE или *GXine* для Gnome. Если вам нужна мощь – берите *Xine*, поскольку ничто не сравнится с его функциональностью, особенно на старых машинах. *Mplayer* тоже весьма полезен, и не только благодаря превосходному модулю видеовывода, но и благодаря тому, что он является чудесной утилитой конверсии.

Но нам-то подавай *VLC*. Этот плейер просто взывает к нашим чувствам компоманов. У него один из лучших пользовательских интерфейсов, к которому можно пристроить чрезвычайно забавный скин. Он до того самостоятелен, что не использует общих движков, и его реакция намного лучше, чем у других плейеров в нашем Сравнении. И он способен передавать видео по сети. У других приложений имеются сходные функции, но у *VLC* эта функция работает по-настоящему, и она стоит дополнительных усилий. Конечно, ею не всегда понадобится воспользоваться немедленно, она пригодится только в том случае, когда нет доступа к настоящему DVD-плейеру, но все же это круто.

В общем и целом, каждое рассмотренное приложение – значительный шаг вперед по сравнению с тем, что нам предлагали еще пару лет назад. Единственное, к чему можно придраться – ни одно из приложений не предназначено для незатейливого воспроизведения DVD. Все они желают быть хозяевами известной им вселенной, а не скромным работягой – DVD-плейером. Конечно, хотелось бы увидеть эквивалент *WinDVD* или *PowerDVD* для Windows. Но пусть это вас не расстраивает. Пару лет назад мы даже мечтать не могли о таких хороших программах, а сейчас вы можете наслаждаться просмотром DVD почти в любой системе Linux, так что мы все – в выигрыше. LXF

Следующая ступень дистанционного управления – *VLC* можно управлять из web-интерфейса.

VLC 8/10

ДЛЯ ВАС

Какими из этих приложений вы пользуетесь постоянно? Может, вы считаете, что мы что-нибудь пропустили? Поделитесь мыслями: letters@linuxformat.ru

СВОДНАЯ ТАБЛИЦА ВОЗМОЖНОСТЕЙ

	GXine	Kaffeine	KMPlayer	MPlayer	Totem	VLC	Xine
Пропуск рекламы				✓		✓	
Микширование аудио каналов	✓	✓	✓	✓	✓	✓	✓
Меню DVD	✓	✓	✓		✓	✓	✓
Закладки	✓	✓	✓		✓		
Фильтры	✓	✓			✓		✓
Переходы	✓	✓		✓	✓	✓	✓
Субтитры	✓	✓	✓	✓	✓	✓	✓
Списки воспроизведения	✓	✓	✓		✓	✓	
LIRC	✓	✓	✓	✓	✓		✓
Вывод на экран (OSD)	✓	✓	✓	✓	✓	✓	✓
Потоковое вещание			✓			✓	

ВЫБИРАЕМ

HTTP ПРОКСИ-СЕРВЕР ДЛЯ ЗАЩИТЫ КОРПОРАТИВНОЙ СЕТИ

Сергей Иванов анализирует существующие Unix-решения с открытым исходным кодом.

В настоящее время количество угроз безопасности в сети Интернет растет с каждым годом. Это особенно касается пользователей Windows. Прямой доступ к Интернет с рабочих станций Windows стал небезопасен, поэтому администраторы сетей часто организуют доступ через посредника соединений.

Для предоставления доступа посреднику необходимо решить следующие задачи: авторизация пользователей, установка виртуального соединения между клиентом и сервером, защита клиента от атак из сети Интернет, балансировка нагрузки, аудит трафика, фильтрация информации, кэширование полученной информации. В качестве посредника может быть использован HTTP прокси-сервер. Связь с Интернетом происходит исключительно через него, все остальные соединения запрещаются администратором сети. Таким образом обеспечивается защита пользователей от внешних атак, а также возможность контролировать информацию через единую точку доступа.

Между клиентом локальной сети и сервером прокси-сервер организует односторонний виртуальный канал, исключая возможность создания обратного соединения со стороны сервера. Таким образом, если предположить, что соединения внутри локальной сети безопасны, отпадает необходимость в настройке брандмауэра на пользовательских машинах. Выход наружу через единую точку облегчает реализацию политики безопасности, хотя и налагает дополнительные требования к отказоустойчивости и быстродействию прокси-сервера.

Рассмотрим возникающие задачи и способы их решения.

Авторизация пользователей

Первая задача, которая возникает перед администратором прокси, – авторизация и идентификация пользователей для выхода в Интернет через прокси. Поддержка прокси таких схем авторизации как NTLM, MSNT, SMB, LDAP, существенно облегчает решение поставленной задачи. Идентификация особенно важна, когда необходимо решить вопросы аудита и фильтрации трафика.

Интеграция в сеть

При интеграции прокси в корпоративную сеть необходимо определиться со способом взаимодействия прокси и пользователей. Существует два способа интеграции.

Первый, и, наверное, самый простой, – когда пользователь самостоятельно конфигурирует свои сетевые настройки для использования прокси. Конечно, простым он кажется только на первый взгляд, поскольку когда сеть очень большая, работа по настройке ложится либо на администратора сети, либо на самих пользователей. Существуют различные варианты оптимизации такой настройки, но в любом случае часть работы все равно должны выполнять как пользователи, так и администратор.

Второй вариант – использование механизма «прозрачного прокси» («transparent proxy» или, более точно – перехватывающего прокси – «intercepting proxy»). Реализация данного механизма заключается в автоматическом перенаправлении HTTP-запросов пользователей на прокси-сервер. Перенаправление осуществляется на шлюзе, который соединяет внутреннюю сеть с Интернетом. В прокси требуется поддержка определения оригинального адреса пользователя, отправившего запрос. Часто шлюз и прокси физически представляют один сервер, особенно в небольших сетях. Поскольку данный вариант интеграции не требует настроек пользовательской машины, достигается эффект «прозрачности» доступа пользователя во внешнюю сеть.

Таким образом, «прозрачный» режим требует поддержки и со стороны прокси-сервера, и со стороны шлюза, но позволяет достичь простоты в сопровождении.

Если в качестве шлюза используется сервер, поддерживающий протокол WCCP, есть смысл организовать «прозрачное» проксирование с использованием этого протокола. Протокол позволяет отслеживать статус прокси-сервера, и если сервер стал недоступен, организовать перенаправление иначе. Это позволит избежать отказа в обслуживании (DoS).

Кэширование трафика и балансировка нагрузки

Поддержка кэширования входящего трафика – одно из важных требований к прокси-серверу. Запрашиваемые пользователем объекты сохраняются в постоянном кэше на диске, и в случае если запрос происходит повторно, объект извлекается из кэша, а не скачивается из Интернета. Это позволяет экономить входящий трафик, что часто имеет большое значение для компании. Когда сеть объединяет большое количество рабочих станций, нагрузка на прокси-сервер многократно увеличивается. В этом случае применяется каскадирование прокси-серверов, и точка доступа во внешнюю сеть получается распределенной за счет увеличения количества серверов.

Для взаимодействия между ними используется протокол ICP, который позволяет быстро определить наличие кэшированной копии объекта на прокси-сервере. ICP используется только для обмена контрольной информацией; обмен данными между прокси-серверами происходит через

протокол HTTP. За счет равномерного распределения данных между разными прокси-серверами достигается балансировка нагрузки. Равномерность такого распределения обеспечивается специальными алгоритмами и протоколами, такими как CARP и Round-robin. Применение CARP обеспечивает более равномерное распределение нагрузки и оптимизирует поиск кэшированных копий объектов на прокси-серверах, поскольку использует hash-алгоритм, основанный на URL.

Аудит и фильтрация информации

Контроль информации, которая циркулирует между клиентом и сервером, – типичная задача, которая стоит перед компанией. Для такого контроля необходимо решить задачи аудита и фильтрации информации. Для реализации этих задач был разработан протокол ICAP (Internet Content Adaptation Protocol). Протокол позволяет осуществлять модификацию HTTP-запросов и HTTP-ответов, а поскольку это единственные запросы, которые циркулируют между прокси и пользователями, полный контроль над трафиком обеспечен. С помощью ICAP-протокола можно производить различного рода действия над HTTP-трафиком, такие как антивирусная проверка, блокирование спама, предоставление доступа к приватным ресурсам, блокирование доступа к порно-ресурсам, учет трафика и многое другое. Одно из самых популярных применений ICAP – фильтрация вирусов. Чтобы получить картину в целом, рассмотрим принцип работы ICAP. Прокси-сервер выступает в качестве ICAP-клиента, который взаимодействует с ICAP-сервером, используя протокол ICAP. Конечный пользователь взаимодействует с серверами через прокси, используя протокол HTTP. Таким образом, поддержка ICAP-протокола конечному пользователю не нужна, и адаптация HTTP-сообщений происходит для него «прозрачно».

Для адаптации ICAP клиент передает HTTP-сообщения, инкапсулированные в ICAP-протокол ICAP-серверу, который может производить различные действия, вплоть до полной замены содержимого. До создания этого протокола не существовало общего метода для внедрения в прокси-сервер модуля контроля информации.

Что же дальше?

Вооружившись общими сведениями о прокси, можно перейти к рассмотрению конкретных примеров. Из прокси-серверов с открытым кодом наиболее популярны Delegate, Oops и Squid. Для получения наиболее полной картины, целесообразно сравнить эти продукты не только с точки зрения поддерживаемых возможностей, но и с точки зрения архитектуры. Ведь именно архитектура определяет дальнейшую судьбу продукта, как в области применения, так и в области развития. Итак, рассмотрим каждый из них.

Delegate

Многоцелевой прокси-сервер.

- **ВЕРСИЯ:** 9.1 • **WEB:** http://www.delegate.org/
- **Цена:** бесплатно для применения в небольших сетях (до 500 пользователей)

Многоцелевой прокси-сервер, работающий с различными TCP-, UDP-протоколами, такими как HTTP, HTTPS, FTP, NNTP, SMTP, SOCKS, IMAP, ICP и т. д. *Delegate* работает как прокси уровня приложения (application-level), который пропускает через себя различные протоколы, производя необходимые изменения в данных. Также *Delegate* может быть использован как прокси для передачи любых данных между клиентом и сервером по протоколам TCP, UDP. Являясь прокси уровня приложения, *Delegate* обеспечивает виртуальное представление ресурсов, расположенных на различных серверах. В случаях HTTP, FTP, NNTP, *Delegate* может выступать самостоятельным сервером. Поддерживается фильтрация данных. Для фильтрации используется собственный интерфейс CFI (Common Filtering Interface). В качестве фильтров могут быть использованы как внутренние средства *Delegate*, так и обычные внешние приложения (sed, awk...), работающие со стандартным вводом/выводом. В случае когда *Delegate* выступает в качестве HTTP прокси-сервера, для хранения объектов может быть использован постоянный кэш, размещенный на диске. Поддержка протокола ICP позволяет эффективнее взаимодействовать с другими кэш-серверами. Авторизация пользователей, если она необходима, происходит через PAM и собственную службу авторизации DGAuth.

Для ограничения загрузки сервера поддерживается ограничение по количеству соединений, тайм-ауту. Если взглянуть на *Delegate*, создается ощущение, что он представляет собой гигантский запутанный конструктор, из которого можно смастерить практически любой прокси-сервис.

Тем не менее функциональности для кэширующего прокси недостаточно.

Прозрачное HTTP-проксирование поддерживается не полностью, поскольку реализация опирается только на данные «Host:» HTTP-заголовка, а он не всегда присутствует. Не поддерживаются протоколы WCCP, CARP. Вместо стандартного протокола ICAP поддерживается свой собственный интерфейс фильтрации, что затрудняет встраивание стандартных сервисов фильтрации и аудита. Не поддерживаются схемы авторизации NTLM, MSNT, что затрудняет интеграцию в сеть, построенную на технологии Windows.

Скажем несколько слов об архитектуре *Delegate*. Сервер каждого протокола работает в отдельном процессе и при каждом соединении с пользователем порождает новый процесс.

Для пересылки информации по каналу данных в протоколе FTP создаются дополнительные процессы. Фильтры также создают дополнительные процессы для запуска внешних программ фильтрации. Скорость работы таких фильтров оставляет желать лучшего. Когда количество клиентов достигает хотя бы тысячи, создается большое количество процессов, что ведет к неразумному расходу ресурсов и большой загрузке системы.

Несмотря на все перечисленные недостатки, *Delegate* может быть весьма полезен в небольших сетях для решения простых задач, поскольку весьма прост в конфигурации. В таких сетях загрузка прокси будет небольшой, и проблемы с производительностью не возникнут.

Oops!

Альтернатива Squid.

- **ВЕРСИЯ:** 1.5.23 • **WEB:** http://www.oops-cache.org/
- **Цена:** бесплатно, по лицензии GPL

Это полноценный HTTP прокси-сервер, который был разработан в качестве альтернативы *Squid*. Поддерживаются протоколы HTTP, FTP, SSL-туннелирование.

Основными целями, которые преследовали разработчики *Oops*!, были: модульность, высокое быстродействие, простота конфигурирования, длительная бесперебойная работа. *Oops*! – это кэширующий прокси-сервер.

Хранение кэшированных объектов реализовано в одном или нескольких файлах. Каждый файл содержит внутри себя файловую систему. Для взаимодействия с кэш-серверами поддерживается протокол ICP. *Oops*! полноценно поддерживает режим прозрачного проксирования, поскольку опирается не только на данные HTTP-заголовка, но и на NAT. Поддерживается протокол WCCP, что позволяет оптимально использовать режим прозрачного прокси. Для авторизации пользователей поддерживается PAM, LDAP, стандартный файл паролей. Также доступна авторизация через MySQL, PostgreSQL. Для сбора статистики реализована поддержка протокола NetFlow. Контроль доступа осуществляется через набор правил, определяемых через ACL.

Помимо очевидных плюсов существуют и минусы. Отсутствует поддержка протокола CARP, в результате балансировка нагрузки между различными серверами не может быть выполнена с максимальной эффективностью. Хотя вариантов авторизации довольно много, не поддерживаются схемы NTLM,

»

« Oops!

(окончание. начало на с.33)

MSNT. Отсутствует поддержка протокола ICAP, что затрудняет решение задач фильтрации. Благодаря поддержке NetFLow и ACL, задачи аудита могут быть решены, но эффективное решение для фильтрации трафика не реализовано. После рассмотрения основных плюсов и минусов с точки зрения поддерживаемых возможностей перейдем в архитектуре *Oops*!

В *Oops*! используется модель, где каждое соединение обрабатывается отдельным потоком.

Если с точки зрения простоты программирования этот подход имееат свои преимущества, то с точки зрения производительности он далеко не оптимален. В общем-то это одно из основных отличий от архитектуры *Squid*, который вообще не требует использования потоков и использует однопроцессную модель. За счет выбранной модели разработчики *Oops*! стремились обеспечить лучшее масштабирование на SMP-системах.

Выбранная архитектура была не самая удачной, поскольку создание потока на каждый выполняемый запрос нерационально с точки зрения использования системных ресурсов.

На SMP-системах для обеспечения масштабируемости достаточно создавать пул потоков и распределять задачи между ними, используя неблокирующие операции ввода/вывода.

Возвращаясь к использованию потоков вообще, хотелось бы заметить, что они и по сей день работают не на всех Unix-системах одинаково эффективно, а на момент создания *Oops*! реализацию потоков можно было назвать полноценной только в Solaris, для которой он, собственно, и разрабатывался.

При большой нагрузке у *Oops*! возникают серьезные проблемы с производительностью на большинстве Unix-систем, а зачастую он попросту зависает. Таким образом, основные требования для сервера – бесперебойная работа и высокая производительность – не выполнены. Тем не менее *Oops*! вполне можно использовать в небольших сетях благодаря простоте настройки.

Squid 2.5

Стандарт де-факто.

- **ВЕРСИЯ:** 2.5.SATBLE13 • **WEB:** http://www.squid-cache.org/
- **Цена:** бесплатно, по лицензии GPL

Вот уже много лет *Squid* является наиболее популярным высокопроизводительным HTTP прокси-сервером на Unix-системах. Поддерживаются такие протоколы, как HTTP, SSL, FTP, Gopher. *Squid* может работать как в режиме кэширующего прокси, так и в режиме HTTP-акселератора. Хранилище кэшированных объектов реализовано в виде модульной виртуальной файловой системы. Используемые файловые системы определяются конфигурацией *Squid*. Поддерживаются следующие системы: ufs, diskd, aufs, coss. Список модулей может быть легко расширен. По умолчанию используется ufs. Модули diskd и aufs используются для оптимизации работы с файловой системой при большой нагрузке. При необходимости распределения нагрузки кэш-сервера могут каскадироваться, взаимодействуя по протоколу ICP.

Поддержка Round-robin и CARP обеспечивает балансировку нагрузки между серверами. Поддерживается полноценная работа в режиме «прозрачный прокси». Поддержка протокола WCCP позволяет использовать прозрачный режим наиболее эффективно. Довольно распространенный протокол SNMP для поддержки удаленного контроля и управления также поддерживается *Squid*. Это позволяет отслеживать его поведение с единой консоли администратора.

Для авторизации пользователей поддерживаются NTLM, MSNT, LDAP, SMB, что позволяет легко интегрировать *Squid* в сеть Microsoft. Одна из удобных возможностей, которую поддерживает *Squid* – ограничение пропускной способности по настраиваемым критериями. Такая функциональность позволяет ограничить использование канала в Интернет для определенных групп пользователей. Особенно важной такая функциональность становится в условиях экономии трафика, поскольку позволяет устанавливать свою пропускную способность для разных типов трафика.

Для управления доступом и сервисами используется мощное средство – ACL. На основании данных HTTP-заголовка и URL, используя ACL, можно выполнить статическую фильтрацию трафика. К сожалению, с динамической фильтрацией трафика дела обстоят несколько хуже. Кроме redirector, других возможностей для внешнего контроля данных в *Squid* не предусмотрено. В экспериментальной версии *Squid* 2.5 поддерживается протокол ICAP, но его реализация обладает рядом серьезных недостатков:

- не поддерживается фильтрация FTP-трафика. Для реализации такой фильтрации необходимо объединять два прокси-сервера;
- не поддерживаются постоянные соединения с ICAP-сервером. Таким образом, каждый запрос клиента порождает новое соединение с ICAP-сервером, что не оптимально при большой нагрузке;
- неполная поддержка ICAP RFC.

Возможно, именно эти недостатки помешали включить поддержку ICAP в стабильную ветку.

Squid довольно мощное гибкое средство и предоставляет большие возможности по его оптимизации. В вопросах оптимизации может помочь статистика реального времени, которая накапливается в процессе работы. С помощью CacheManager, входящего в состав *Squid*, ее можно просматривать через Web. Конечно же, это не единственное применение такой статистики. За счет оптимизации производительность *Squid* может быть улучшена, хотя в конфигурации по умолчанию скорость его работы вполне достойная.

Из перечисленных выше продуктов *Squid* – самое высокопроизводительное решение.

Высокое быстродействие и переносимость обеспечивается выбранной архитектурой. *Squid* использует однопроцессную архитектуру, когда один процесс обслуживает все операции ввода/вывода в неблокирующем режиме. Состояния дескрипторов отслеживаются с помощью функций poll, select и т.д. Для оптимизации некоторые виды действий, такие как dns-запросы или операции с диском, могут быть выделены в отдельные процессы. Вспомогательные процессы взаимодействуют с основным через IPC-функции.

Squid не предполагает обязательного использования потоков, чем обеспечивает хорошую переносимость между различными Unix-системами без потери производительности.

Использование потоков в *Squid* является альтернативой оптимизации дисковых операций, для которых часто не поддерживается неблокирующий ввод/вывод. Для SMP-систем это является дополнительным плюсом, так как улучшает масштабируемость.

Сетевая модель, лежащая основе архитектуры *Squid*, – одна их самых оптимальных, хотя и не достаточно учитывает SMP-системы. Распределение обслуживания сетевых соединений между несколькими процессами (потоками) позволило бы улучшить масштабируемость.

На базе *Squid* реализовано довольно много проектов. Некоторые из них перечислены на **http://www.squid-cache.org/products.html**. Это и специализированные устройства, и выделенные сервера. Очевидно, что *Squid* может быть адаптирован под самые различные задачи.

Squid 3.0

Адаптирован под самые различные задачи.

• **ВЕРСИЯ:** 3.0 • **WEB:** http://www.squid-cache.org/

Новая веха в судьбе *Squid*.

Основные отличия от версии 2.5:

- Конвертирование исходного кода из C в C++. Довольно важный шаг, поскольку облегчит сопровождение продукта и расширение функциональности.
- Поддержка ICAP. Поддерживается не только адаптация HTTP-трафика, но и FTP.
- Переработана сетевая часть. Оптимизирована работа со стандартными функциями select, poll. Добавлена поддержка более высокопроизводительных интерфейсов отслеживания состояния дескрипторов: **epoll**, **/dev/poll**, **kqueue**. Поскольку *Squid* обрабатывает все сетевые соединения одним процессом, а эффективность poll/select функций сильно падает при большом числе дескрипторов, данная оптимизация позволяет ускорить работу сетевой части.
- Улучшена поддержка CARP. В ранних версиях поддержка по умолчанию отсутствовала, что создавало неудобства в использовании, поскольку требовалась пересборка *Squid*. В версии 3.0 поддержка CARP присутствует по умолчанию.
- «Прозрачные» и «акселерированные» запросы обрабатываются независимо. В ранних версиях такие запросы были связаны между собой в результате ограничения, накладываемые на «акселерированные», влияли на «прозрачные» запросы, что нелогично. Возникали неоднозначности при авторизации. Авторизация стала доступна только для «акселерированных» запросов.
- Улучшена работа в режиме акселератора. Более удобные и понятные опции конфигурации, повышен уровень безопасности.
- Улучшена поддержка SSL. Добавлена поддержка SSL peers, поддержка сертификатов клиента, поддержка аппаратных SSL-ускорителей при помощи OpenSSL, более защищенная работа с SSL-ключами.
- Переработана поддержка IPC.
- Расширена поддержка ACL. Добавлены новые методы.

Список изменений довольно большой. Вполне понятно, почему разработчики решили сделать эти изменения в новой ветке, а не развить ветку 2.x. Проделана большая работа, которая позволит дать толчок дальнейшему развитию продукта.

Среди добавленных возможностей хотелось бы особенно выделить поддержку протокола ICAP: возможность, которая позволит решать задачи аудита и фильтрации в полном объеме. Хотелось бы отметить, что реализация ICAP в *Squid* 3.0 – не наследие неэффективной экспериментальной версии из ветки 2.5, а новая, более производительная реализация. Поддержка ICAP позволит интегрировать в *Squid* антивирусы и другие внешние сервисы для обработки HTTP-трафика, что сделает сеть более защищенной.

Последняя версия 3.0 уже вышла на финальную стадию стабилизации, вновь добавленные возможности позволят ей стать достойной заменой версии 2.5.

Для подведения итогов объединим полученную информацию в виде таблицы.

СРАВНИТЕЛЬНАЯ ТАБЛИЦА

	ICP	CARP	WCCP	Авторизация	NetFlow	SNMP	Xephem	Сетевая модель	Ограничение полосы пропускания
Delegate	да	а		PAM, DgAuth				процесс на соединение	
Oops!	да	а	да	LDAP, SQL, PAM	да			поток на соединение	
Squid 2.5	да	необходима пересборка	да	NTLM, MSNT, PAM, SMB, LDAP, NSCA ...		да		Один процесс на все соединения	да
Squid 3.0	да	да	да	NTLM, MSNT, PAM, SMB, LDAP, NSCA ...		да	да	Один процесс на все соединения	да

ВЫБИРАЙТЕ!

Hot Picks

Лучшие на планете новинки открытого ПО!

НА ДИСКЕ

Майк Сондерс
Майк не может усидеть с одним MBR больше 30 минут.

Сотни программ с открытым кодом создаются и обновляются каждый месяц, поэтому среди них непросто найти достойную использования. В разделе «Наш выбор» мы просеиваем все последние версии различных программ и отделяем зёрна от плевел, отбирая только самое лучшее. И ежемесячно самое впечатляющее приложение получает наш приз!
Все программы нашего обзора доступны на прилагаемом к журналу диске. Если вы заметили программу, которую стоит рассмотреть, черкните письмо на **mike.saunders@futurenet.co.uk**.

HOTPICKS В ЭТОМ РАЗДЕЛЕ

Angry, Drunken Dwarves 33
GLibrary 34
Haiku 32
Keep 31
Kiso 30
KTechlab 31
Peekko 34
PouetChess 33
Revelation 35
Zoinks 35

ОБРАТИТЕ ВНИМАНИЕ НА ПОБЕДИТЕЛЯ HOTPICKS

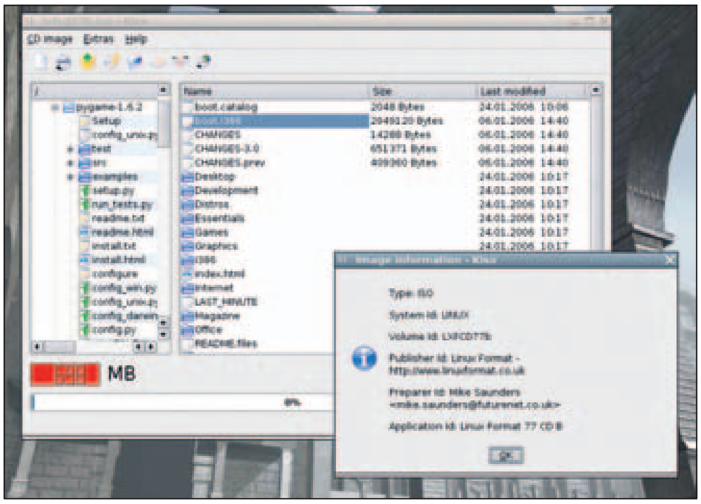
В нашем разделе **HotPicks**… все заслуживает внимания, но ежемесячно мы награждаем один проект за его выдающиеся свойства. Выбирается ну самый лучший!

HOTTEST PICK!

СОЗДАНИЕ ОБРАЗОВ ДИСКОВ

Kiso

• **ВЕРСИЯ** 0.8.3 • **WEB** http://kiso.sourceforge.net



Кланяемся Кодерам Kiso за Кличку Короче чем *KompactDiskKonstructor*…

Имя *Kiso* построено по обычной формуле KDE-приложений: берете функцию программы, ставите перед ней букву K – и готово дело! *Kiso* – графическая оболочка создания ISO-образов дисков для записи на CD или DVD. ISO-образы могут содержать дистрибутивы Linux, резервные копии личных данных и другое-прочее – стандарт ISO9660C, определяющий способ представления данных на диске, поддерживается большинством операционных систем.

Под Linux немало графических утилит, выполняющих те же функции, что и *Kiso*, а также всяких консольных программ (*mKisofs* и иже с ней), любимых пользователями за гибкость и применение в скриптах. Цель *Kiso* – сделать процесс создания образов дисков простым и быстрым, сохранив гибкость на радость продвинутым пользователям. Программу написал на C++ Стефан Ганс (Stephan Gans) из Киля (Kiel), Германия.

> «ЦЕЛЬ KISO – СДЕЛАТЬ СОЗДАНИЕ ISO-ОБРАЗОВ ПРОСТЫМ И БЫСТРЫМ.»

Kiso разработана для KDE 3, поэтому для ее установки из исходных текстов понадобятся пакеты разработки (обычно **qt-devel** и **kdelibsdevel**). Кроме того, программа требует библиотеку *libcdio*, отвечающую за чтение CD-ROM – она имеется в папке *Kiso* на нашем диске. Далее установите программу, используя обычные команды **./configure**, **make** и **make install** (эти команды выполните от имени **root**). Если программа не добавилось в меню, просто наберите *Kiso* в терминале.

При первом запуске *Kiso* появится окно, в котором большинство опций будет недоступно, и потребует приступить к работе. Но если нажать на иконку Создать образ, *Kiso* спросит, хотите ли вы создать новый образ диска, и выдвинет лоток вашего CD-дисковода, а когда вы послушно вставите болванку, ровно ничего не сделает. Поэтому на первый вопрос лучше ответить «нет». Тогда в главном окне слева появится древовидная структура вашей файловой системы. Теперь не составит труда перетащить в правую часть окна те файлы, из которых нужно составить новый диск. Здесь можно создать и новые директории, обычным образом – через правую кнопку мыши.

Дискотека

LED-индикатор внизу показывает количество памяти, занимаемое текущим образом диска, а индикатор прогресса показывает, сколько времени займет процесс генерации диска. Этого вполне достаточно для изготовления резервных копий на CD или DVD, однако *Kiso* способна на большее.

Через меню Extras можно указать файловый образ, чтобы получившийся диск был загрузочным. Например, при создании дистрибутивов Linux можно указать загрузочный образ дискеты.

Дополнительно можно установить метку тома и имя издателя диска (по умолчанию эти поля заполняются именем файла ISO-образа). Классная особенность программы – возможность преобразования ISO в другие форматы, например, NRG, используемый в *Nero Burning ROM*. Вы можете даже прочитать существующий ISO образ, зашифровать его с помощью *mcrypt* и записать одним щелчком мыши.

В настоящий момент у программы нет онлайн-документации, а набор ее опций ограничен. Однако эти недостатки не критичны. Хотя в *Kiso* отсутствуют полезные командные опции в стиле *mKisofs*, для большинства обыденных операций создания дисков и резервных копий программа подходит идеально.

МЕНЕДЖЕР РЕЗЕРВНЫХ КОПИЙ

Keep

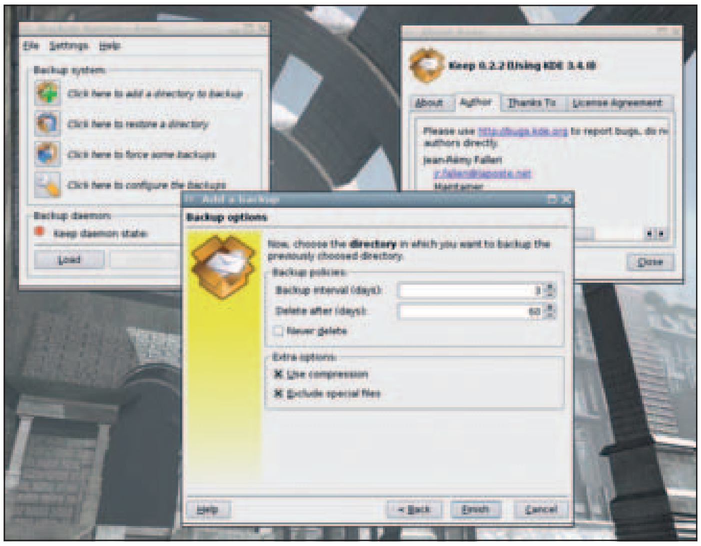
• **ВЕРСИЯ** 0.2.2 • **WEB** http://jr.falleri.free.fr/keep

Большинство из нас сталкивается с задачей резервного копирования. Конечно, стабильность Linux и устойчивость файловых систем снижают риск потери данных из-за выхода из строя системы или сбоев электроэнергии, но плохие CD/DVD-R диски, хромающее оборудование и поврежденные архивы могут доставить немало головной боли. Выход из этой ситуации – регулярное и автоматизированное создание резервных копий, очень кропотливый и трудоемкий процесс, особенно при работе в консоли. Однако этот процесс можно существенно упростить – спасибо понятному интерфейсу *Keep*!

Keep разработана для среды KDE, поэтому для того, чтобы собрать ее из исходных текстов, понадобятся пакеты разработки KDE и Qt (обычно **kdelibs-devel** и **qt-devel**). В качестве движка программа использует разработанную на Python утилиту *rdiff-backup*, вы найдете ее в папке *Keep* на прилагаемом диске.

Разработчики *Keep* позаботились о вашем удобстве. При щелчке на любой иконке появится мастер, под руководством которого вы выберете директорию, предназначенную для копирования, затем директорию, где требуется сохранить резервную копию, а также политику копирования: временные интервалы, через которые будут создаваться копии, и заказ на сжатие полученных копий. Если вы захотите переопределить установки, это можно сделать из главного окна программы.

> «ЗАПРОСТО ДЕЛАЕТ МНОГО РЕЗЕРВНЫХ КОПИЙ В УКАЗАННЫЕ МОМЕНТЫ.»



Простой мастер установки *Keep* идеален для новичков в Linux.

Keep подстраивается под ваши нужды: вы можете архивировать одну директорию или, столь же просто, большое их число, причем каждая будет иметь свои свойства и интервалы архивации. Фоновый демон отслеживает текущую конфигурацию, автоматически производя архивацию согласно вашим правилам. Правда, хотелось бы большей свободы действий, например, устанавливать интервалы копирования в часах или месяцах, а не только от 1 до 30 дней, а также использовать для сжатия не один *Gzip*, а и другие архиваторы, например, набирающий популярность 7-Zip.

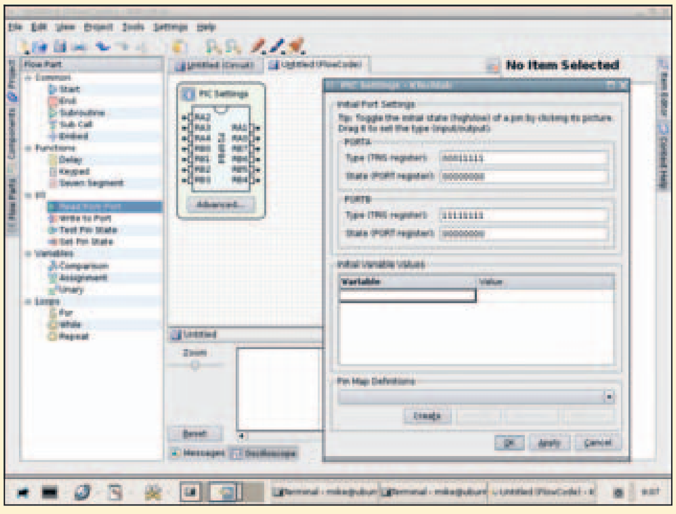
Недостатки, впрочем, ничуть не умаляют достоинств. Программа проста в обращении, не удручает пользователя непонятным техноязом – короче, это отличный инструмент KDE для обеспечения сохранности данных.

IDE ДЛЯ ИНТЕГРАЛЬНЫХ СХЕМ

KTechlab

• **ВЕРСИЯ** 0.3 • **WEB** www.ktechlab.org

***KTechlab* – это «интегрированная** среда разработки для электрических схем и микроконтроллеров». Другими словами, она позволяет располагать виртуальные компоненты на плате, моделировать их поведение и анализировать результаты. Как и во многих других средах, поддерживается разработка в стиле drag&drop. *KTechlab* разработана для среды KDE и дополнительно может использовать пакет **gputils** для сборки программ, работающих на PIC-контроллерах.



Даже ранняя версия *KTechlab 0.3* не уступит иному 1.0.

Две панели главного окна содержат множество закладок и иконок. Расположены они очень удобно и не вызывают дискомфорта. Подсказка предложит создать проект на основе электрической схемы, блок-схемы или программы на C/Ассемблере для поддерживаемого микроконтроллера PIC14. В режиме построения электросхем доступен древовидный список компонентов, которые можно перетащить на панель редактирования: это источники питания, резисторы, лампы и интегральные схемы. Редактор компонентов позволяет задать их свойства, тут поможет превосходная контекстная справка, лаконично объясняющая назначение каждого компонента.

Режим FlowCode, используемый для графического построения PIC программ, имеет вид редактора блок-схем с циклами, условиями и подпрограммами. Используя простые инструменты, можно добавлять в документ линии, примитивы и текст. Затем блок-схема FlowCode преобразуется в язык высокого уровня Microbe, а тот, в свою очередь, преобразуется в ассемблерный код при помощи встроенного компилятора. Если у вас установлен *gpasm* (из пакета **gputils**), то можно сгенерировать двоичный код программы. В программу встроен отладчик для исследования PIC программы – и на Ассемблере, и в режиме FlowCode.

Для релиза с номером 0.3 документация *KTechlab* необычно обильна: онлайн-руководство описывает режимы редактирования и позволяет привыкнуть к интерфейсу программы, описание языка Microb сопровождается примерами и справочником по синтаксису. И контекстная справка, и руководство обрадуют пользователей. Конечно, *KTechlab* не предмет первой необходимости для каждого, но те, кто проектирует и программирует микроконтроллеры, оценят ее по достоинству.

ОПЕРАЦИОННАЯ СИСТЕМА

Haiku

HOTTEST PICK!

• **ВЕРСИЯ** 2006-01-23 • **WEB** www.haiku-os.org

Ранее известная как OpenBeOS, Haiku является клоном Be Operating System, разработанной фирмой Be Inc в 90-х годах прошлого века. Для своей эпохи BeOS выглядела впечатляюще: легкая в обращении, интенсивно использующая многопоточность, она работала и реагировала на действия пользователя гораздо быстрее, чем Windows и Linux, и, как следствие, покорила сердца многих фанатов на рынке мультимедиа.

Однако доминирование Windows на рынке усиливалось, а Linux оставалась системой для умников. Борьба за существование закончилась для BeOS переходом в собственность Palm. Нельзя отрицать и давления со стороны Microsoft – Be Inc даже подавала в суд на непорядочные методы конкуренции. До суда, однако, не дошло: Microsoft заплатила 23 миллиона долларов фирме, практически дышащей на ладан.

Перед самой кончиной Be Inc, небольшая группа разработчиков, ободренная успехом Linux, решила почти с нуля создать бесплатный и открытый клон BeOS. За основу взяли открытое ядро NewOS, написанное Трэвисом Гейзельбрехтом (Travis Geiselbrecht) из Be, и использовали обширную документацию проекта BeOS для написания требуемых составляющих, подобно тому как GNU Project занимался созданием свободных аналогов компонентов Unix. В настоящее время Haiku (в этой поэтической форме сообщалось об ошибках в приложениях BeOS) набирает обороты и находится на грани превращения в отличную ОС.

Дело движется

Цель первого главного релиза Haiku – достичь двоичной совместимости с последней версией BeOS, R5, и когда выйдет Haiku R1, вы сможете запускать большинство программ с сайта BeOS **www.bebits.com**. Пока еще живы коммерческие приложения BeOS, например, офисный пакет *Gobe Productive* и полупрофессиональные средства работы с мультимедиа.

Пользователи Linux будут как дома в структуре файловой системы BeOS и Haiku: здесь присутствуют символьные ссылки, придающие сходство с Unix, а также поддержка стандартов POSIX, позволяющая запросто перенесить традиционные Unix-программы, например, *XEmacs* и Perl. В BeOS даже имеется порт *Firefox*, и можно надеяться, что скоро он заработает в Haiku.

Набор приложений, поставляемый с Haiku, пока ограничен небольшими утилитами – текстовым редактором и программой записи звука (в **/Haiku/beos/apps**), а также утилитами конфигурации, обеспечивающими системе ее прославленное дружелюбие. Разработчики Haiku достигли производительности исходной ОС и почти полностью сохранили ее внешний вид. Последнее время с Haiku занималась группа добровольцев, но для ускорения проекта завели штатного сотрудника – Акселя Дорфлера (Axel Dorfler), самого плодовитого из них.

Многие люди недоумевают: зачем возиться с Haiku вместо того, чтобы довести Linux до такого же высокого уровня? Дело в том, что в случае прекращения процесса разработки Linux (например, по юридическим причинам) важно иметь альтернативную открытую ОС как запасной вариант.

Что ждет Haiku в будущем? Немецкая фирма YellowTab сейчас готовит расширение последней версии BeOS, называемой Zeta, права на которую были приобретены у Palm. Haiku пока не является полноценной ОС, но YellowTab использует в своем продукте некоторые части открытого проекта, например, *OpenTracker*.

Haiku – продукт многообещающий: быстрая, легкая, стабильная и дружелюбная настольная ОС казалась миражом миру ПК, но Haiku, в конечном итоге, может воплотить мечту и стать альтернативой имеющимся настольным системам. Пока что Haiku просто интересный курьез для изучения, но в ней кроется огромный потенциал. Пожелаем удачи команде разработчиков.

ПРОБУЕМ НА ВКУС

HAIKU

Haiku можно запустить в эмуляторе Qemu, он есть на нашем диске в разделе System. Установите прекомпилированную версию или соберите его из исходных текстов. Затем запустите виртуальный диск с Haiku с помощью

```
qemu haiku.image
```

Учтите, что запустить Haiku с модулем ускорения Qemu нельзя, а как только появится окно Qemu, необходимо щелкнуть внутри него, чтобы в Haiku заработала мышь. После редкостно быстрой загрузки (чем и славилась BeOS) вы окажетесь в окне интерпретатора команд. Зрелище не слишком впечатляет? Так это только основа ОС – инсталляторы и изысканные мастера добавятся потом. А сейчас вы можете запустить Deskbar (аналог панели задач Windows) и Tracker (менеджер файлов):

```
/Haiku/beos/system/Deskbar &
/Haiku/beos/system/Tracker &
```

Теперь вверху справа вы видите меню программ и список запущенных приложений, а в левом верхнем углу – иконки ваших дисков. Интерфейс BeOS менее стильный, чем у KDE, Vista или OS X, но он освежающе-просторный и позволяет продуктивно работать. Легко заметить, что основные компоненты ОС быстро взрослеют: ядро способно загружаться на настоящем оборудовании ПК или эмуляторе; стабильно работает appServer, отвечающий за управление GUI; присутствуют также многие другие подсистемы.

Интерфейс пользователя Haiku не перегружен графикой, зато очень быстр и приятен в работе.

ШАХМАТЫ

PouetChess

• **ВЕРСИЯ** 0.1 • **WEB** http://pouetchess.sourceforge.net

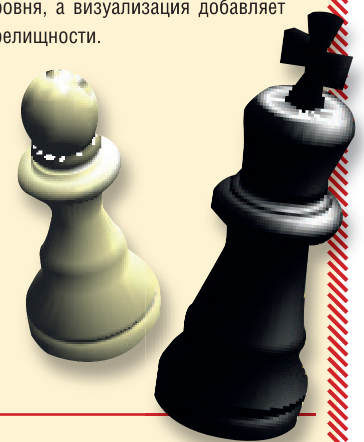
Wikipedia завершает описание шахмат фразой: «Число возможных партий намного превышает количество атомов во Вселенной» – пусть Ludo знает свое место! [Прим. пер.: Ludo – настольная игра для двух или четырех человек, в которой необходимо провести 4 фишки от начальной точки к конечной. Количество ходов определяется броском игральных костей. Более подробно об этой игре – на **http://en.wikipedia.org/wiki/Ludo (board game)**.] Компьютерные шахматы всегда были предметом глубокого изучения, ибо это игра логики и стратегии – тут исключены факторы случайности и везения. В Linux пользователи играют с программой *GNUChess* – невзрачный на вид, но сильный игрок. Задача *PouetChess* – добавить в игру изюминку с помощью красивой 3D-графики.

Чтобы собрать *PouetChess*, понадобятся *SDL* и *SDL_mixer*, имеющиеся в большинстве дистрибутивов (и в разделе Essentials на нашем диске), а также OpenGL. Процедура установки не совсем обычная: необходимо ввести **make** в директории, куда вы распаковали архив, а после компиляции надо перейти в директорию **bin** и запустить *PouetChess* оттуда.

PouetChess поддерживает и одиночный режим, и режим для двух игроков. В одиночном режиме предусмотрены 3 уровня интеллекта компьютера. Уровень и максимальное время обдумывания хода компьютером проставляются в файле **config.ini**. Естественно, все зависит от вашего процессора. Автор рекомендует использовать CPU с частотой от 500 МГц, чтобы программа успевала делать хороший ход за терпимое время.

Ха, размен коня и слона – равноценный, годится. Ход конем и вправду похож на галоп.

Главный козырь *PouetChess* – приятная графика. Пуристы шахмат, конечно, заявят, что графика значения не имеет: мол, сложный игровой процесс поглощает вас целиком. И все равно для глаз приятнее объемные фигуры с тенями и такая домашняя деревянная доска (можно даже менять угол зрения, правой кнопкой мыши), чем унылая плоская двухцветная диаграмма. В целом, *PouetChess* идеальна для новичков и игроков среднего уровня, а визуализация добавляет зрелищности.



ГОЛОВОЛОМКА

Angry, Drunken Dwarves

• **ВЕРСИЯ** 0.91 • **WEB** http://www.sacredchao.net/~piman/angrydd/

Говорят, не суди о книге по обложке. А можно ли судить об игре по названию? Название *Angry, Drunken Dwarves* («Злые пьяные гномы») подходит дурацкой стрелялке для приставки Mega Drive… но вы удивитесь: это головоломка. Итак, похоже, что нельзя.

Нет, в игре нет ни упоения насилием, ни злобных интриг. Тематика добавлена просто для колорита, иначе получился бы рядовой клон игр с падающими кубиками. Сдержанный юмор разработчиков не утомителен и оживляет игру.

В соответствии с сюжетом, некто очень сильно вас разозлил. Кто – в общем, неважно. И вы решили излить агрессию на гномов, кидаясь в них драгоценными камнями. Чтобы дойти до жизни такой, вам понадобится как минимум установить Python 2.3 и *Pygame 1.6.2* и набрать заветные команды **make check** и **make install** (от лица суперпользователя). После этого игра появится в **/usr/local/games/Angryadd**.

ADD – типичный представитель своего класса: цветные кубики падают сверху, а ваша задача – собирать их в группы по четыре, после чего группа превращается… в один большой камень. Камни разрешается разбивать; осколки исчезают и возникают на экране противника, мешая ему организовывать собственные группы камней.

Кстати о противнике: в *ADD* можно играть и против человека, и против компьютера, который ведет себя как мелкое двуногое существо в подпитии. Игровой процесс сопровождается веселыми биографическими текстами и картинками. Отработанная графика и продуманный звуковой фон свидетельствуют о профессионализме – чувствуется, что игра сделана добротно, она развлекает и при этом не надоедает.

NB: Чтобы получить удовольствие от игры, не обязательно предварительно принимать спиртное или раздражаться!

Борода-Лопатой: бравый безбашенный безобразник.

КАТАЛОГИЗАТОР КНИГ

GLibrary

- **ВЕРСИЯ** 1.0.1
- **WEB** www.glibrary.svx.pl/phpfusion_en/news.php

Будучи фанатом Unix и/или Linux, вы наверняка накопили по ним столько книг, что хватит на целую полку, а то и не одну. Разобраться во множестве томов – непростая задача: пусть книги расставлены в алфавитном порядке или рассортированы по категориям, но если их более 30, тщетно пытаться вспомнить, в которой содержится срочно нужная вам информация. Тут-то и пригодится *GLibrary*. Это скоростной каталогизатор книг, сравнительно небольшого размера; его высокую производительность обеспечивает движок *SQLite*. Для компиляции программы необходимо иметь *GTK* 2.6, *Libglade* 2.0 и *SQLite* 3.0 или более поздние версии.

Главное окно *GLibrary* прямо-таки голое, всего с парой кнопок и полем для поиска нужной книги справа вверху. Супер-графики подобным программам не требуется, и все-таки было бы неплохо иметь еще несколько кнопок и небольшое приветствие при загрузке программы. Добавляя книгу, можно сопроводить ее одну или несколькими записями. Окно Book Details содержит поля названия, жанра и издателя книги, а также прочие детали: ISBN, число страниц и даже имя переводчика. Книгу можно снабдить комментарием.

После создания записей в главном окне программы возникает список названий книг, с типом и подробностями. Режим управления базой данных позволяет редактировать авторов, жанры, языки и другие данные – увы, при добавлении новой книги придется все вводить вручную (лучше бы был выпадающий список). Благодаря *SQLite* поиск супербыстр – гораздо быстрее, чем в XML или обычных текстовых файлах. Подобные программы мы уже рассматривали в нашей рубрике.

GLibrary не бальзам для глаз (разве что серый цвет – ваш любимый). Не помешали бы дополнительные опции, например, показ обложки книги (аудиоприложения давно уж без этого не обходятся). Но если вы просто хотите упорядочить свое собрание книг, не отвлекаясь на избыток возможностей, то эта программа – хороший помощник.

Вид у *Glibrary* неказистый, зато интерфейс работает очень эффективно.

ЧАТ-РАСШИРЕНИЕ FIREFOX

Peekko

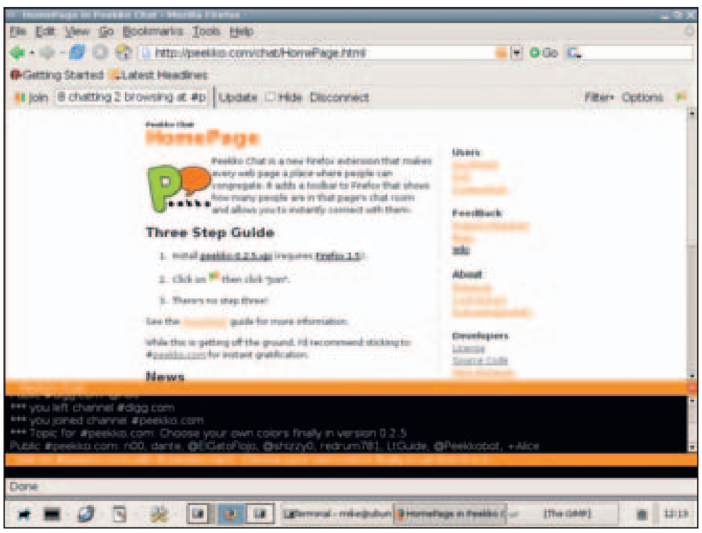
- **ВЕРСИЯ** 0.2.5 • **WEB** http://peekko.com

Огромное число расширений для *Firefox* – несомненно, его сильная сторона. С течением времени появляется все больше новых идей. Небольшое расширение *Peekko* Chat позволяет пользователям перемыть косточки сайту, куда они все забрели одновременно. Чтобы установить плагин, зайдите на сайт **http://Peekko.com** и щелкните на ссылке **Peekko-0.2.5.xpi** (вам понадобится *Firefox* 1.5).

После перезапуска *Firefox* вы увидите вверху новую панель, меняющуюся по мере перехода с сайта на сайт. Например, на сайте *LXF* программа сообщит вам нечто похожее на «3 общаются, 7 глазеют». Первое число показывает, сколько народу используют расширение в действии, а второе – сколько владельцев *Peekko* просто зашли на страницу. Нажмите кнопку Join, и внизу браузера появится новая панель: цветовая схема с черным и оранжевым нам не кажется привлекательной, но зато панель занимает мало места.

А вы не заметили ничего знакомого на приведенном снимке экрана? Да, *Peekko*, по существу, миниатюрный IRC-клиент, который соединяется с собственным *Peekko*-сервером, а название канала образуется от имени посещаемого сайта. То есть если вы читаете Slashdot, то входите на канал #slashdot.org. Ясно, что модератор не в силах обслужить такую прорву IRC-каналов, поскольку теоретически канал полагается каждому домену в мире; авторы *Peekko* должны позаботиться, чтобы каналы не забивались рекламой и взаимной руганью. Да и над графикой поработать, уж очень она сейчас незатейлива.

Насколько популярно расширение? Ну, *Peekko* достаточно свежая разработка, и пока нечасто натыкаешься на компанию беседующих, хотя на многих технических сайтах дискуссии уже ведутся вовсю. Ситуация изменится, когда Peeko (и *Firefox*) наберет популярность. А сейчас это маленькая система, умело раскрутившая существующую технологию (IRC). Если она соберет достаточно поклонников и они дадут ей шанс, Сеть получит еще одно измерение. Web 3.0, мы здесь!



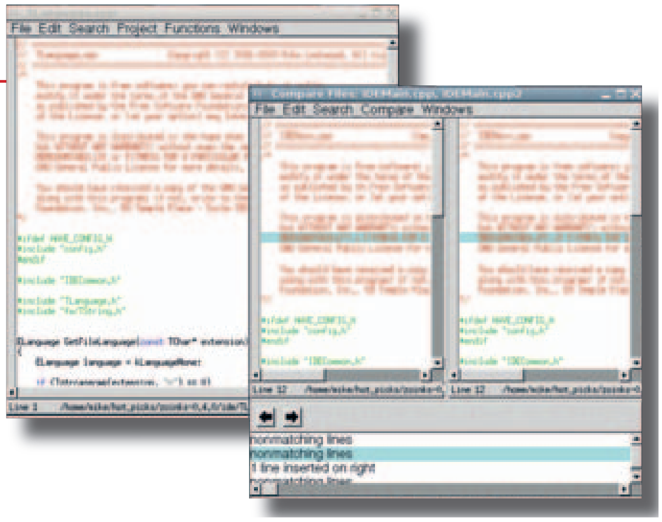
Вверху на панели показано, сколько людей в данный момент общается, а нижнее окно – мини-IRC клиент.

HotPicks ПОВТОРНЫЙ ВИЗИТ

ТЕКСТОВЫЙ РЕДАКТОР

Zoinks

- **ВЕРСИЯ** 0.4.0
- **WEB** http://zoinks.mikelockwood.com



Инструмент *Zoinks* для сравнения файлов; различия подсвечиваются.

В *LXF40* мы рассказали о *Zoinks*, легком редакторе, ориентированном на разработчиков. Его производительность и стабильность произвели на нас приятное впечатление. Мы рекомендовали его как альтернативу монструозным *Emacs* и *Vim*, потому что с ним очень удобно быстро редактировать код. *Zoinks* написан на C++. С тех пор, как мы рассматривали версию 0.2.8, он продолжал потихоньку развиваться, правда, сохраняя несколько неуместный свой GUI.

В *Zoinks*, как полагается, встроены подсветка синтаксиса C, C++, Java, HTML и TeX, а также дополнительные функции редактирования HTML, например, вставка тэгов. На новый уровень, почти что уровень среды разработки, его поднимают возможности работы с проектом: вы можете определить директорию компоновки, команды компиляции и отладчик для группы исходных файлов, а затем запустить **Make** из меню или комбинацией клавиш. Для *KDevelop* это не конкурент, но штука полезная.

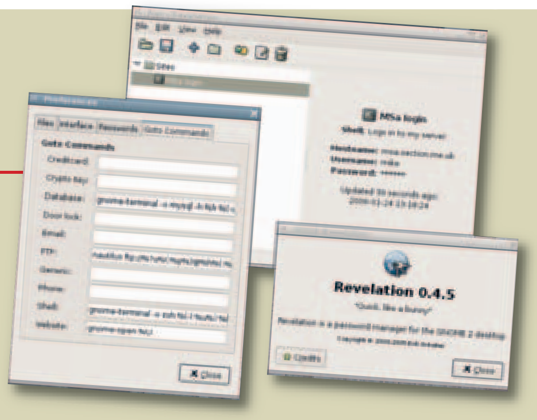
С прошлого раза в программе сделаны следующие изменения: появилось удобное окошко сравнения двух файлов или директорий; встроен просмотр руководства; подсвечивается синтаксис языка Ruby; можно открыть файл на нужной строке, указав ее при запуске команды; много мелких исправлений и обновлений.

Zoinks местами выглядит немного топорно. Но если вы не фанат крупных приложений *Gtk* и *Qt*, то оцените скорость и легкость этой программы.

ХРАНИТЕЛЬ ПАРОЛЕЙ

Revelation

- **ВЕРСИЯ** 0.4.5
- **WEB** http://oss.codepoet.no/revelation



Интерфейс немного изменился, но *Revelation* все еще аккуратна и компактна.

Не так давно мы рассказывали о *Revelation* (в *LXF59*). Мы были счастливы найти добротную программу, действительно помогающую хранить, организовывать и управлять паролями, а не валить их в кучу. Движок программы написан на Python, а оболочка рассчитана на Gnome. *Revelation* позволяет хранить информацию о доступе к вашим банковским счетам и паролях SSH и электронной почты, и даже генерировать пароли автоматически, специально для тех, кто не в силах придумать нечто более оригинальное, чем имя любимой собаки.

Программа позволяет рассортировать пароли по категориям. Например, пароли, относящиеся к всемирной паутине, можно хранить в одной папке, а PIN-коды – в другой. Полезная возможность – оценка паролей, в форме словесной рекомендации, например, «слишком короткий» или «недостаточно уникальный» и т.п. Записи можно ассоциировать с программами. Так, например, двойной щелчок по учетной FTP-записи откроет, по умолчанию, *Nautilus*.

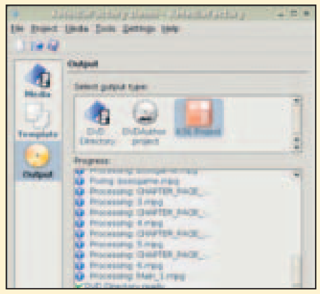
Среди новых функций, появившихся с ноября 2004 года, интересна поддержка импортирования файлов с паролями из других программ: *Password Safe*, *GPass* и *Password Gorilla*. Добавился апплет к Gnome, для быстрого просмотра нужной записи, а также автоматическое блокирование файла с данными после определенного периода бездеятельности, повышающее безопасность. Наконец, добавилась поддержка экспорта в виде обычного текста – для тех, кто доверяет своему принтеру…

Revelation и во время нашего прошлого обзора представляла собой отличный продукт. Новые возможности будут более чем востребованы, а интеграция с Gnome – подарок пользователям, пользующимся данной средой. Кому требуется хранить более пяти паролей, немедля приступайте к установке!

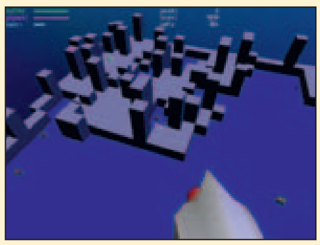
НОВЫЕ РЕЛИЗЫ

Новые и обновленные программы, заслуживающие внимания…

- ***pyvm 1.1*** Виртуальная машина Python **http://students.ceid.upatras.gr/~sxanth/pyvm**
- ***QA Tools 1.2*** Создание снапшотов **http://lintouch.org/static/qa-tools.html**
- ***FONTpage 1.0*** Просмотр шрифтов **www.pcbypaul.com/software/FONTpage.html**
- ***mved 0.1*** Переименование файлов (для параноиков) **http://raf.org/mved**
- ***KMediaFactory 0.5.0*** Создание интерфейсов к DVD **http://susku.pyhaselka.fi/damu/software**



- ***KXDocker 0.40 RC*** Переписанный док для KDE **http://tinyurl.com/393ah**
- ***Stendhal 0.43*** Онлайновая многопользовательская RPG **http://arianne.sourceforge.net**
- ***AsltHappens 0.2*** Монитор производительности сети **www.abrahams.co.nz/asithappens**
- ***MMR 1.6.0*** Простой почтовый клиент a la Pine **www.devolution.com/~slouken/projects/mmr**
- ***Grml 0.6*** Live CD дистрибутив на базе Knoppix для системных администраторов **http://grml.org**
- ***Enemy Lines 3 1.1*** 3D-стрелялка **http://raum1.memebot.com/enemylines/enemylines3.html**



- ***Sound Converter 0.8.3*** Аудиоинструмент Gnome **http://soundconverter.berlios.de**
- ***InitNG 0.5.3*** Супер-быстрая загрузка Linux **http://initng.thinktux.net**
- ***PortForward 1.3*** Создание защищенных тоннелей **http://brisa.homelinux.net/casa/portforward**
- ***Makepack 0.0.1*** Создание пакетов для Slackware **http://makepack.sourceforge.net**
- ***Liquid Weather ++ 9.2*** Тема для SuperKaramba **http://liquidweather.net**
- ***Wiclear 0.9*** Wiki сервер с регулировкой оформления **http://wiclear.free.fr/?en/Wiclear**

Что такое... JINGLE?

Открытое VoIP-расширение для быстрого обмена сообщениями? **Энди Ченнел** неразлучен с Jingle.

НА ДИСКЕ

>> «Ну есть ли время лучше...» Погодите. Рождество вроде кончилось?

Ну и что? Jingle можно использовать круглый год. Это новый набор библиотек, его можно встроить в вашу любимую программу обмена сообщениями и совершать звонки через Интернет, используя компьютер с Linux. [обыгрывается рождественская песенка «Jingle Bells», – прим. ред.]

>> Так это просто *Skype*?

Нет, но вопрос не такой глупый, как может сперва показаться. *Skype* – это приложение VoIP (передача голоса по IP), доступное на разных платформах. Оно позволяет общаться с помощью компьютеров, а также совершать звонки на обычные телефоны. Проблема в том, что *Skype* – бесплатная, но не открытая в терминах сообщества Open Source программа. Вдобавок она принадлежит eBay, и это чревато прекращением поддержки Linux-версии и всякими нежелательными сюрпризами – шпионами, рекламными баннерами и прочими ужасами.

>> А как насчет *Gizmo*, *Google Talk* и других *Skype*-подобных приложений? Они ведь основаны на открытых стандартах?

Это правда, потому-то Jingle так актуален. Jingle Signalling и его компаньон Jingle Audio – новые расширения XMPP (Extensible Messaging and Presence Protocol), являющегося основой технологии для Jabber. Jingle Audio пожинает лавры, а Jingle Signalling вкалывает, заменяя в Jabber текущий протокол передачи (TINS) на новую систему, способную устанавливать соединение между абонентами и управлять им, а также беспрепятственно работать через брандмауэры и проводить маршрутизацию для обеспечения наилучшего качества звонка.

Что касается Jingle Audio – этот способен работать с любым аудиопотоком, хотя спроектирован для RTP (протокол для приложений реального времени). Оба расширения были созданы совместными усилиями Jabber Software Foundation (JSF) и программистами поисковика-корифея Google. Расширения легли в основу VoIP-решения *Google Talk*.

Более того, чтобы продвинуть технологию в массы и расширить выбор приложений, фирма Google выпустила свою библиотеку *libjingle* по лицензии BSD, а также опубликовала API для *Google Talk*. Теперь любой желающий может сделать совместимым с *Google Talk* свое IM- или VoIP-приложение.

Сравните текущее положение с недавним прошлым, когда большие игроки – Microsoft, Yahoo и AOL – всячески ограждали тайной свои программы, чтобы приковать пользователей к своим системам. Наступила эпоха перемен!

>> Звучит красиво. Ну и что будет?

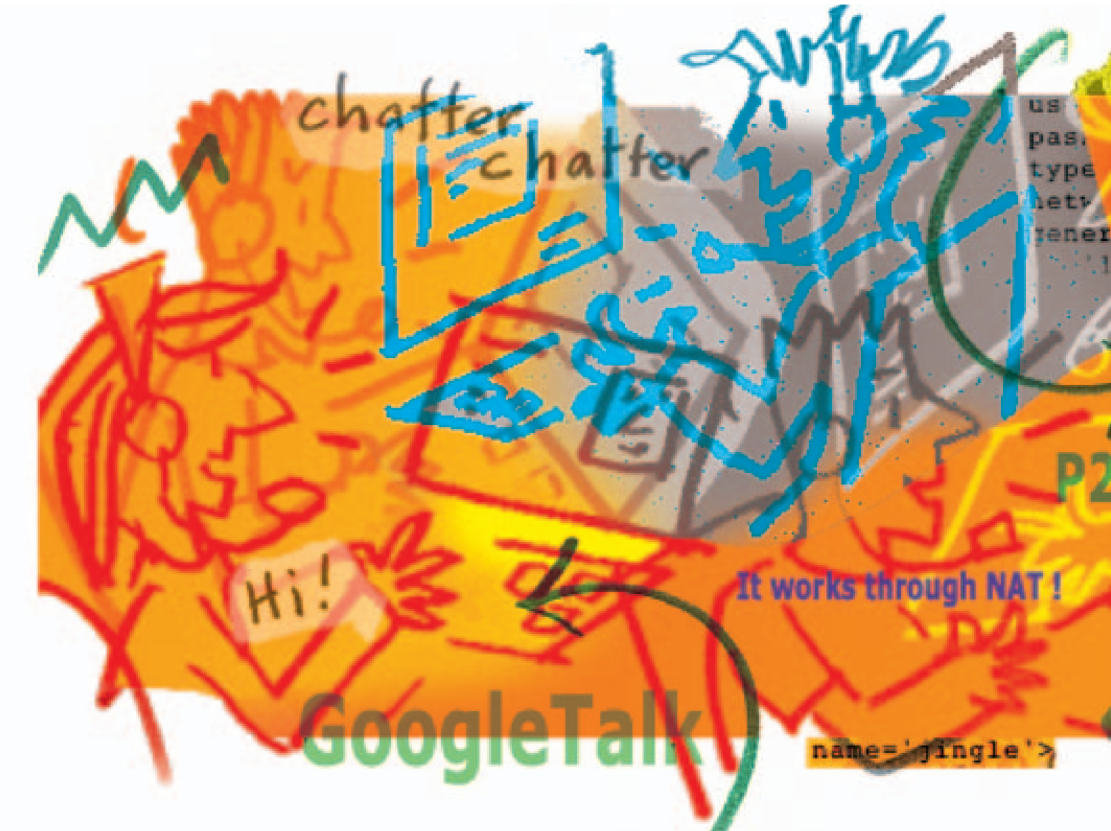
В конечном счете, VoIP должен пойти тем же путем, что и электронная почта с телефоном – отказаться от привязки к конкретной сети. Но есть и сиюминутные изменения. Представьте, что вы разработчик, решивший составить конкуренцию *Skype* своим VoIP-приложением. У вас есть две реальных альтернативы: создать новую инфраструктуру и убедить достаточное число людей скачать и установить вашу программу, чтобы проект был жизнеспособным; или же использовать *libjingle* и дать возможность пользователям звонить своим товарищам в *Google Talk*, используя ваш клиент.

>> Беда в том, что я отнюдь не разработчик.

Jingle представляет интерес также и для пользователей, особенно для тех, кто предпочитает Linux другим операционным системам. Возьмем Google: применяя программное обеспечение с открытым исходным кодом практически во всех сферах своей деятельности, эта фирма, тем не менее, не особо стремится выпускать под Linux собственные приложения – *Picasa*, *Google Earth* и, как вы уже догадались, *Google Talk*. В любом случае, Google заинтересован в привлечении к *Google Talk* возможно большего числа абонентов, а потому и упростил подключение к своей сети пользователей, не работающих под Windows, а также предусмотрел сотрудничество с другими сетями. До появления *libjingle*, такое было возможно лишь в некоторых приложениях, например, *Gaim* и *Kopete* (частично основанном на Jabber). Теперь вы можете скачивать приложения, способные подключаться к *Google Talk*, не затрудняясь поисками компьютера с Windows. Ну что, ура?

>> Ура! Похоже, что Google прибавила себе пользователей, фактически не выпустив готового продукта для своей ключевой ОС.

«КАЖДЫЙ МОЖЕТ ДОБИТЬСЯ СОВМЕСТИМОСТИ С GOOGLE TALK ДЛЯ СВОИХ VOIP ПРОГРАММ.»



Какие приложения поддерживают данную возможность?

Большинство команд разработчиков Linux-программ обмена сообщениями уже выразили намерения принять Jingle, и на момент выхода данной статьи, видимо, уже будут альфа-, бета- или даже полнофункциональные версии, использующие новые преимущества. Один из первых проектов, внедривших их в жизнь – *Gizmo*, запущенный основателем Linspire Майклом Робертсоном (Michael Robertson), чтобы не отставать от *Skype*. Как и *Skype*, *Gizmo* доступен для Linux, Mac OS X и Windows, но он появился на рынке слишком поздно и не сумел завоевать достаточное число пользователей – так что принятие Jingle может выправить ситуацию. Разработчики *Gizmo* уже наладили контакт с пользователями *Google Talk* и работают над добавлением возможности VoIP – хотя пока что умалчивают о сроке ее появления.

Gaim 2.0 также находится в стадии бета. Однако в имеющемся списке изменений не упоминается ни Google, ни Jingle – даже невзирая на то, что ведущий разработчик *Gaim* Шон Иган [Sean Egan] теперь работает в Google.

В свою очередь, недавнее сотрудничество Google и AOL означает, что *AIM* скоро (возможно) будет взаимодействовать с сервисами Google; ходит много слухов о том, что Apple включит *libjingle* в следующую версию iChat; а Yahoo! добавит такую возможность в своем Messenger. Я уверен, что такая ситуация не понравится программным монополистам.

>> Кто же первый предложит Jingle?

Я не собираюсь гадать, но думаю, что рабочую версию уже имеет *Kopete*. У кое-каких продвинутых пользователей она даже работает (совершались звонки), однако сборка из исходных текстов не проходит гладко – ну, то есть, она вообще не собирается под Unsermake, которым KDE заменило Automake. Так что если хотите собрать *Kopete* с поддержкой Jingle, убедитесь, что оно не использует Unsermake.

>> Где я могу взять *Kopete* с поддержкой Jingle?

Вы можете воспользоваться *SVN* и получить текущую разрабатываемую ветку. Попробуйте

```
svn co -N svn://anonsvn.kde.org/home/kde/branches/KDE/3.5/kdenetwork
cd kdenetwork
svn co svn://anonsvn.kde.org/home/kde/branches/KDE/3.5/kde-common/
admin
svn co svn://anonsvn.kde.org/home/kde/branches/work/Kopete/dev-0.12/
Kopete
```

На момент прочтения, уже может появиться ветка поновее – не поленитесь проверить на сайте *Kopete* (**http://Kopete.kde.org**).

Определившись с версией, отключите *Unsermake* и запустите команды конфигурации:

```
export UNSERMAKE=no
make -f Makefile.cvs
./configure --enable-debug --enable-jingle
cd Kopete
make
su -c «make install»
```

>> Надо ли позаботиться еще о каких-нибудь зависимостях?

Вам понадобится несколько программ, включая *ORTP* (в настоящий момент это должна быть версия 0.7.1), *Expat* (анализатор XML), *Glib 2.0* и Speex (версии 1.1.6 либо 1.0.5).

>> Сложновато. А нет ли другого способа?

Не совсем – во всяком случае, не сейчас. Появление скомпилированных версий в различных дистрибутивах Linux не за горами. VoIP из тех программ, на которые технари набрасываются с великим энтузиазмом. Они будут сутками напролет просиживать за компьютерами, чтобы поскорее предъявить нам готовый продукт.

>> Независимая от сети передача голоса по IP звучит заманчиво, а что еще умеет делать Jingle?

Как я уже отметил, вся суть Jingle Signalling – работа на основе клиент-клиент (P2P), через наилучший маршрут между двумя подключенными компьютерами. Благодаря этому он может, со слов Майка Джазаери [Mike Jazaeri] из Google, использоваться для видеоконференций, обмена файлами, удаленного доступа к приложениям и совместной работы.

>> Стоп-стоп. Вы сказали, удаленного доступа к приложениям?

Да, и интересный вопрос, не следует ли считать это технологией несанкционированного доступа. Представьте будущее, в котором пользователи вводят в Google свои учетные записи, используя браузер – так же, как они это делают для обмена сообщениями, электронной почты или RSS – и получают немедленный доступ к широкому классу приложений и ко всем своим документам по сети. Причем процесс еще ускоряется выбором оптимального маршрута передачи данных по замечательным волоконным проводам, которые, по слухам, приобретает Google. Это будет чудесный рождественский подарок… жаль, что на дворе еще только апрель. LXF

Mono-глоты

Linux Format встретился с **Эдом Дамбиллом** (Edd Dumbill) и **Нилом Борнштейном** (Niel Bornstein), авторами недавно вышедшей книги о прелестях Mono.

LXF ИНТЕРВЬЮ

Споры по поводу Mono наэлектризовали свободное сообщество. Должны ли мы писать программы, которые целиком и полностью зависят от изобретения Microsoft? Не нарвемся ли мы на патентные иски? Долго ли еще играть в догонялки с инновациями Microsoft? Некоторые считают, что инновации .NET слишком серьезны, чтобы занять позицию «изобретено не здесь» и проигнорировать их, и полагают, что Mono (открытая реализация .NET) и C# могли бы стать наилучшим выбором для будущих разработок.

Эд Дамбилл и Нил Борнштейн – двое из вышеупомянутых некоторых; они соавторы книги «Mono: Блокнот разработчика» (Mono: A Developer's Notebook), ставшей настольным лабораторным путеводителем для программистов, которые делают на пути Mono первые шаги. Мы поймали писательский дуэт на OSCon, чтобы узнать, что подвигло их на создание этого труда и почему они избегают C++, и, конечно, расспросить о книге.

Linux Format: **Вы, парни, работаете с Linux много лет. Как вы вышли на Mono, и чем занимались все годы до него?**

Эд Дамбилл (ЭД): Я работал над проектом рабочего стола Gnome и тесно соприкасался с вещами, которые приходилось делать ребятам из Ximian. На Mono вышел потому, что это позволило мне добиться большей продуктивности в написании Gnome-приложений, совершив скачок на C# из C.

Так что для меня дело было не в кросс-платформности, а просто Mono – отличная среда для работы. Я принял участие в проекте, который запустил [основатель Ximian и Mono] Нат Фридман (Nat Friedman) под названием *Dashboard*, на какое-то время проект завис, а потом превратился в поисковый инструмент под названием *Beagle*. Вот так я и пришел к Mono.

Нил Борнштейн (НБ): Что до меня, я перешел от Java к .NET под Windows, и меня всегда интересовали возможности Open Source в этой области. Эд как-то пришел и сказал, что хочет написать книгу [о Mono], ну и я вроде как заинтересовался. Я пришел к Mono от книги, а не наоборот – к книге от Mono.

LXF: **До сих пор кое-кто не доверяет Mono, так как боится проблем с патентами. Вы, ребята, с Mono уже сравнительно давно. Вы знали об угрозе патентных исков, и все-таки решились нырнуть в него с головой?**

ЭД: Люди, которым я доверяю, Нат и Мигель [де Икаса (de Icaza), основатель Mono], потратили на Mono очень много времени, для меня лично это в некотором смысле достаточно надежная гарантия. Я тогда не представлял никакую корпорацию, просто был программистом. И потом, каждый раз, когда вы пишете программу, вы почти наверняка нарушаете чей-нибудь патент. Но даже если конкретно такого патента нет, при желании можно найти похожий, и вас все равно потащат в суд.

LXF: **Но это же не считается умышленным нарушением?**

ЭД: Нет, но проблема в том, что если вы – мелкая рыбешка, а патентные условия сформулированы достаточно расплывчато, причем в США в большинстве случаев так и есть, то за вами начнут охоту. И еще до суда у вас не останется денег на борьбу. Проблемы с патентами в большей степени касаются той части Mono, которая не является стандартом ECMA [European Computer Manufacturers Association, Европейская Ассоциация Производителей Компьютеров], то есть эмулирующей Windows, а я никак не использую эту часть платформы. Windows Forms, ASP .NET...

НБ: Microsoft опубликовала эти стандарты, то есть передала их в ECMA, и теперь они общественные – так почему бы и не воспользоваться ими? А остальное… кто знает, что будет? Мне ничто не мешает использовать это остальное, и не помешало бы мне предлагать другим его использовать, если надо.

LXF: **Перейдя на Mono, какие главные преимущества вы нашли для себя?**

ЭД: Главное различие – в переходе с C на C# и написании программ для рабочего стола Gnome. Этим занималось немало людей в среде Gnome с помощью по-настоящему интересных и полезных приложений, которые сейчас быстро развиваются.

Одно из основных преимуществ перехода на зрелый объектно-ориентированный язык – большое количество возможностей. Настольные приложения Mono сегодня просто первоклассные, в них есть наследование, события и прочее. Все это, конечно, есть и в Java, но в C# вы чувствуете себя преемником программистов C. Я никогда не мог по-настоящему привыкнуть к Java, потому что в нем слишком много ограничений на то, чем я занимался.

НБ: Да еще в C# имеется много прима-нок, которых не было – а некоторых и сейчас нет – в Java. Так что мне, когда я отошел от Java, было очень приятно продолжить работать на C# (мне он понравился еще под Windows) и использовать его для Linux, Mac и всех других платформ.

ЭД: Еще одно серьезное преимущество – вы не теряете свои старые наработки на C из-за всяких P/invoke. Можно выполнять «родной» код из программы C# и не писать все эти оболочки и тому подобное, а при работе в Python, скорее всего, пришлось бы задействовать SWIG. И это был бы долгий-долгий обходной путь. Вот вам и преимущество.

Во многих языках скриптов и динамических языках вы все время налетаете на препятствие, потому что не нашлась требуемая библиотека – а для молодых языков это весьма частая ситуация. Так что здесь мы имеем отличный выход.

LXF: **Вы сказали, что переход на C# дает вам наследование, а оно есть также и в C++. Разве перейти на C++ не так же легко?**

ЭД: По-моему, надо быть мазохистом, чтобы перейти на C++. Есть причины, по которым проект Gnome этого не сделал. Такой переход влечет множество тягот, а мне их не надо. Я думаю, это дело личного убеждения.

И потом, намного разумнее работать с языком, который впитал опыт использования C, C++, Java. Такой язык берет все хорошее и выбрасывает все плохое, потому что это плохое путается под ногами. Ну, вы знаете, все эти случайности с файлами заголовков, все эти штуки из C, C++… Одна большая головная боль. Нужны были прототипы, по два места для вашего класса.

НБ: Да, грустно. Я совсем забросил указатели. Пользоваться [в Mono] ими можно, но я не вижу для этого причин, если вы пишете портируемый код и не используете унаследованных библиотек. Я не писал на C уже лет пять или шесть.

ЭД: Я думаю, управление памятью – это отличная вещь. Работа в среде управляемого кода.

LXF: **Как сказал Мигель: «В твоей жизни наступает момент, когда ты** >>

TOP TRUMPS COURTESY: WINNING MOVES UK LTD

Визитка LXF

Эд Дамбилл

Редактор по общим вопросам в O'Reilly Network, также работает над ПО Bluetooth для Debian и пишет для Gnome. Он – бывший редактор XML.com и председатель Европейской конференции XML (XML Europe conference)

ВОЗРАСТ	31
НАЦИОНАЛЬНОСТЬ	Британец
ИСПОЛЬЗУЕТ UNIX	9
ЯЗЫКОВ ПРОГРАММИРОВАНИЯ	10
КОЛИЧЕСТВО ПК	7
ДНЕВНАЯ НОРМА КОФЕ	3 чашки
ПАР САНДАЛИЙ	1

ПРЯМАЯ РЕЧЬ «Игра Mono в догонялки с Microsoft не слишком вдохновляет.»

<< **понимаешь, что слишком большую ее часть потратил на создание собственной системы сборки мусора».**

ЭД: Да, пора начать доверять кому-то еще, и мы как раз подошли к этому моменту.

НБ: Я считаю, что память и процессорное время достаточно дешевы, чтобы позволить запустить сборку мусора. Сейчас не надо подстраивать свой код так долго, как раньше.

LXF: **Вы, значит, думаете, что если придется программировать *GTK*, вы прямиком двинетесь к Mono? Или вы скажете: «С тут подойдет лучше»?**

ЭД: Не думаю, что кто-либо в здравом уме будет и дальше работать на C. Основная причина, почему изрядная часть Gnome по-прежнему на C – потому что с этим согласилось большинство народа. Точнее, все сошлись на том, что хотят использовать язык высокого уровня, [но] какой – Python? Или C#? Или новый classpath Java? Я думаю, что C выбран просто благодаря его фундаментальным библиотекам. Но попытайтесь создавать объекты в C или *GTK*, и очень скоро увидите, как это мучительно.

Я думаю, что достоинства есть и у Mono, и у Python. Возможно, Python быстрее по части создания прототипов, но зато типобезопасность C# создает уверенность в том, что ваша программа ни с того ни с сего не сломается. Сейчас реально существует среднее звено – с Boo и IronPython, мы на той стадии, когда имеются достойные интерпретируемые языки, способные запускаться в той же среде. Я бы с удовольствием использовал их и для создания прототипов, не такой уж я упертый.

LXF: **Участвовали ли вы в попытках ввести в проект Gnome другие языки высокого уровня?**

ЭД: Ну, в какой-то мере меня втянули в дебаты, в первый раз это было около года назад. Дебаты всех утомили, но ни к какому решению не привели. По-прежнему пребываем в состоянии неопределенности. Все участники проекта Gnome работают на разные корпорации. У Red Hat есть эта их здоровая штуковина на Java для работы с classpath и так далее. Парни из Canonical в восторге от Python – и они вовсе даже не спешат его покинуть.

ДАМБИЛЛ – ПРО МОНО

> «Засучите рукава – и за код, причем как можно быстрее. Просто скачайте пару проектов и пошарьте в исходниках.»

LXF: **Novell, Red Hat и Canonical: три разных фирмы, три разных языка!**

ЭД: Ну и что? У DVD большая емкость; в наши дни у людей увеличилась ширина канала. Надо будет загрузить три виртуальные машины на три среды выполнения и создать мосты, – создадим. Если это – цена создания достойных приложений для Gnome, то мы готовы ее заплатить. Не проблема. Я думаю, именно в этом направлении и следует идти – пусть каждый делает свою работу, а в конце концов объединимся.

Приложение Gnome определятся не языком, на котором оно написано, а собственным стилем – функция drag&drop работает определенным образом, темы подобраны определенным образом, и так далее. А если говорить о пользователях, то им язык абсолютно безразличен.

По-прежнему спорят, какой будет платформа разработки, но мне думается, что это само по себе ответ – потому что разработчики платформы изберут ту, для которой будет хорошая поддержка, ту, в которой они смогут чувствовать себя комфортно с точки зрения корпоративных интересов. И я думаю, что Mono серьезно опережает других на этом фронте, по причине наличия хорошей поддержки, которая у него имеется, поскольку он создан на плечах платформы Microsoft.

LXF: **Для конечных пользователей Mono – сплошь крутые приложения: *F-Spot* – крут, *Beagle* – тоже крут, *Blam* – ну очень крут! На самом деле, лично мне полезен *iFolder*. Это что, из-за того, что над Mono трудятся только крутые хакеры, и создают только крутые программы?**

НБ: Фирмы, которые уже используют .NET и хотят перейти на свободные платформы, начинают с этой целью использовать Mono. В последние полгода я общался со многими фирмами, интересующимися Mono – они пока еще не перешли на него, но хотят знать: «Что надо сделать? Что это вообще такое? Поможет ли это нам перейти на ASP.NET или писать переносимый код на .NET?» Так что в фирмах движение есть, а вот в продуктах, возможно, оно не столь заметно.

ЭД: Проекты, которые вы упомянули, были созданы для рабочего стола Gnome в Mono. Поймите, немало открытых проектов, которые начинались в .NET, работают и в Mono, поэтому нет особого смысла в написании другой версии. Есть шансы, что большинство проектов либо сразу заработают в Mono, либо потребуют очень незначительной перекомпиляции, чтобы уйти от специфики Windows – вроде обратного слэша в имени файла или еще какой-нибудь ерунды, которую программистам и допускать-то не следует.

И потом, неудивительно, что вы видите только крутые программы: ну что еще вам покажут за пять минут?

LXF: **Нужна ли перекомпиляция для таких ситуаций, как вызов P/invoke, где программист, возможно, захочет вернуться в цикл сообщений Windows?**

ЭД: Дело даже не в этом. Такая проблема существует и в Java: вместо того, чтобы использовать класс пути, чтобы создать ссылку на имя файла, разработчик просто присоединяет обратные слэши. [Потому что разработчики] постоянно прикидывают: сколько будет дисков, где будет храниться результат, как добраться до домашней папки и т.д. Но если вы все делаете правильно, то это не помешает портированию. Мы приводим в

книге несколько основных правил для написания переносимых приложений. Не так уж все сложно, но вы удивитесь: оказывается, на свете полно ленивых разработчиков!

LXF: **Поговорим о книге. Это часть целой серии?**

ЭД: Да, серии «Блокнот Разработчика O'Рейли» (O'Reilly's Developer's Notebook). Я думаю, что она отражает суть работы большинства разработчиков: берешь практический пример, прорабатываешь его, ну, может, лезешь в справочник – и вперед! Менталитет «кода по примеру» присущ большинству программистов.

Книга отражает эту ментальность [на] серии «лабораторных задач». Берем пример, показываем, как он работает, и развиваем его – что будет, если сделать так? А если сделать этак? Или иначе? – а потом подсказываем, где можно найти более подробную информацию о данном предмете. Мы немало потрудились, чтобы обеспечить по каждому вопросу дополнительный материал и чтобы вы смогли двигаться дальше, обучаясь самостоятельно.

НБ: Другое преимущество выбранного формата – возможность добавить немного юмора, а не быть все время мрачным технарем. Там есть заметки на полях, где помещаются всякие шуточки, чтобы чтение было легким и интересным, в противовес сухому техническому чтиву.

ЭД: Писать было очень интересно, и, я полагаю, это сказывается. Если вы писали книгу с удовольствием, то и читать ее приятно.

LXF: **У меня на нее было всего два часа вчера вечером, но по моему впечатлению, она коротковата.**

ЭД: Ну, рискну предположить, что вы за нее не платили, но объем книги соответствует ее цене. Это не энциклопедия…

НБ: … а стартовая площадка.

ЭД: В ней есть действительно хороший и продвинутый материал, так что нельзя сказать, что мы не касались сложных тем: нашей задачей было предоставить достаточно материала, чтобы пример был понятен и вы освоились в этой области, а затем указать направление, куда вам двигаться самостоятельно, если вы хотите добиться большего.

Мы, естественно, представляли себе тех, кто будет читать эту книгу. Самое главное, что это люди, которых не надо кормить с ложечки – это же разработчики, а они способны учиться сами. Книгу надо читать, имея под рукой клавиатуру.

НБ: И книга отнюдь не задумана как исчерпывающая.

БОРНШТЕЙН О ФИРМАХ

«Фирмы желают знать: что такое Mono, и не надо ли им именно это?»

LXF: **Что вы могли бы рассказать о создании для Mono альтернативы *Glade* [визуальный редактор интерфейса для Gnome and *GTK*]?**

ЭД: В процессе разработки находятся два преемника *Glade*. Один из них по некой причине называется Stetic. Предстоит выбор, на каком из них остановиться – один из них пишется на Python, а другой – на C#, возможно, здесь и заключается ответ на вопрос о том, что происходит. Возможно, произойдет интеграция.

MonoDevelop неплохо интегрируется. Недавно появилось дополнение NUnit, так что на боковой панели вы сможете увидеть результаты своих тестов – сколько было успешных, сколько неудачных, а также график, отображающий результаты тестирования во времени. Я думаю, что улучшения еще будут, но для достижения уровня *Visual Studio* или *Eclipse*, например, потребуется пройти долгий путь. Если мы хотим убедить людей перейти с Windows на Linux, нужно, чтобы им было удобно, чтобы у них были все те удобные функции, что и в *Visual Studio*. Достойная программа для создания форм – одна из них.

Есть и другие аспекты. Люди покидают мир Windows с его ужасным абсолютным позиционированием элементов на форме и приходят в *Gtk* с его компоновщиками (box model). Вы можете растягивать окна, как хотите, они легко адаптируются. В принципе, с тем же самым сталкиваются дизайнеры при переходе от печатной работы на HTML: им тоже нужно адаптироваться.

LXF: **Что бы вы посоветовали тем, кто уже использует *Qt* или *GTK* и собирается попробовать Mono? Ну, кроме «Читайте нашу чудную книгу»?**

ЭД: Я бы сказал: засучите рукава и беритесь за работу над кодом, и как можно быстрее. В смысле, когда вы меняете род деятельности, лучшее, что может вам помочь – это практический опыт. Некоторые проекты Mono, вроде *Blam* и *Monopod* (это мой клиент Интернет-вещания), достаточно малы, чтобы не заблудиться. Просто скачайте их – и поройтесь в исходном коде. Попробуйте его и так, и этак, пришлите заплатку. По моему мнению, собственного практического опыта ничто не заменит.

НБ: Читайте MSDN [**http://msdn.microsoft.com**]. Самая лучшая документация по языку C# и библиотекам классов находится на сайте Microsoft. Так что если вы действительно хотите узнать что-то о языке и о библиотеках – вам туда. LXF

Интересуетесь Mono? Пишите на **lxf.letters@futurenet.co.uk**, и мы передадим ваши замечания и вопросы Эду и Нилу.

Эд и Нил беседуют об избавлении от стигматов Microsoft, об *OpenOffice.org*, автоматизируемом из Mono, и о многом другом на www.linuxformat.co.uk/mag/mono.html.

ВТОРОЕ пришествие

В этом введении в Web 2.0, **Ричард Смедли** (Richard Smedley) проигнорирует ореол легенд и вклад венчурных капиталов в поиск серьезных технологий – чего искать? Львиная доля их основана на открытом ПО.

Возможно, вы скажете, что все началось в конце 90-х годов с появлением блогов и wiki. А может, толчком к развитию послужили сообщества типа Kuro5hin и новостные сайты вроде Slashdot, содержимое которых пополняют пользователи. Кстати, разве не старые электронные доски объявлений проложили этот путь? Ладно, согласен, Web 2.0 пророс не из одного-единственного корня. Вообще-то не существует и однозначного определения, что есть Web 2.0. Ясно одно – в нем заинтересовано очень много людей.

Бытует мнение, что интерес этот чрезмерный и вызывается только тем, что технология крутая, оперирует модными словами и притягивает венчурные капиталы. Технический обозреватель Марк Эванс (Mark Evans) пишет: «Энтузиазм по поводу Google и Web 2.0 граничит с неразумным перебором», а это, чтобы вы знали, не слишком хорошо. Одна из причин, по которой *Linux Format* заинтересован в Web 2.0, является применение социального программного обеспечения [Прим. пер.: подробнее можно узнать на странице **http://en.wikipedia.org/wiki/Social_software**], открытых стандартов и технологий. Эта ситуация отличается от сложившейся в конце 90-х годов (эпохи последней большой волны сетевых технологий). Тогда самым модным словом было «портал». Сеть представляла собой публикацию, и миллионы пользователей поделенной между игроками сети были уверены, что Интернет – это Yahoo, CompuServe или America Online.

Территория 2.0

Когда случился обвал дот-комов, выжили только самые крупные порталы, а также ряд популярных сервисов поменьше. Они предоставляли услугу, которая не просто возвращала пользователей на сайт, но и формировала его облик: участие пользователя. Эти сайты (и ПО, на котором они работали – web-логи, wiki и другие) потихоньку росли, поскольку люди начали соображать, зачем все это нужно. Блоги, wiki, синдикация, тэги и социальные закладки – пять самых востребованных инструментов Web 2.0. Возможно, вы знакомы со многими, если не со всеми этими инструментами – по большому счету это все шумиха, а не новая технология Web 2.0.

Однако сейчас идет процесс взаимного оплодотворения – например, браузер Web 2.0 Flock использует сервис Delicious в качестве системы закладок, позволяя пользователям перетаскивать мышью блоги с посещаемых страниц, а также сервис Flickr для публикации фотографий.

Другой сервис, Diigo, позволяет пользователю выделить области страницы, включая изображения, затем написать к ним тэги и сделать закладку. Имеется уникальная возможность написать свои комментарии к выделенным областям текста, которыми можно поделиться с другими пользователями Diigo (а можно и не де-

литься) – идеально для случая, когда вы позабыли, зачем вы добавили в закладки такую большую страницу.

Для всех этих сервисов характерен высокий уровень персонализации – кстати, это один из пунктов определения Web 2.0 – а также чувство сопричастности. Успех сетевых блогов хорошо документирован, а достигнут он благодаря способу использования ссылок – ссылки-то и создают сообщества.

Все на тусовку

К этим пяти инструментам Web 2.0 можно добавить «сборища». По виду это стихийно возникающие богемные вечеринки, а по сути – впечатляющая смесь двух богатых источников данных: интерфейса типа настольного GUI и обработки на стороне клиента. Здесь одним из наиболее часто используемых компонентов является Google Maps, в компании сайта по поиску работы, репертуара театров или других местных событий – для наглядности поиска местоположения.

Большая часть Web 2.0 полагается на Асинхронный JavaScript и XML (Ajax), представляющий обновление старых стандартов представления данных – XHTML и CSS, взаимодействующих через Document Object Model [Прим. пер.: объектная модель документа], при этом для асинхронной передачи данных используется XMLHttpRequest и JavaScript. Эти стандарты и стандартные API и привели к созданию всевозможных «сборищ». XML является ключом к Ajax, но он работает и с другими компонентами Web 2.0. Используя XML, компьютеры могут легко интерпретировать и перемещать информацию по сети, облегчая поиск, каталогизацию и индексацию. XML также вдохнул жизнь в Open Document Format (ODF). Тем временем разработчики Web 2.0 уже давно используют технологии, основанные на XML.

Ни одно явление еще не было так хорошо описано его участниками, как Интернет, поэтому неудивительно, что видный издатель, публицист и рупор Open Source Тим О'Рейли [Tim O'Reilly] стал крестным его новой «коллективной» разновидности: так появился Web 2.0. Сам термин родился в диалоге MediaLive International и O'Reilly о проведении совместных конференций по исследованию этих новых технологий. Первая конференция, организованная O'Reilly/MediaLive и посвященная Web 2.0, прошла в Сан-Франциско в октябре 2004 года. Темой конференции была «сеть как платформа» – вот куда зашла старая парадигма «сеть как публикация».

Многие сайты использовали Ajax для предоставления интерактивных web-сервисов, так что технология удаленных асинхронных скриптов насчитывает уже много лет. Однако повсеместное распространение широкополосного доступа в Интернет кардинально изменило картину. К примеру, на впечатляющем Google Maps (**http://maps.google.com**) благодаря Ajax карта перемещается так плавно, как будто находится на вашем собственном компьютере – а попробуйте воспользоваться модемным соединением: Ajax протащит карту на пару кварталов, и вы надолго зависнете в отсталой «всемирной паутине» прошлых лет, дожидаясь продолжения.

Медленность доступа обусловила закат первых технологий закачки, когда задолго до появления Web 2.0, RSS и Atom, венчурные капиталы поддержали бум дот-комов и фирмы вроде PointCast, доставлявшие потребителю новости прямо на рабочий стол, буквально полезли как грибы. PointCast в 1997 году ухватила свежеизобретенную Microsoft технологию Channel Definition Format для доставки сервисов и сайтов на рабочий стол, а Netscape в своем NetCaster использовала для аналогичных целей Java и HTML. Однако пользователям приходилось ждать по полчаса, чтобы через модем 14.4K бит/с скачались обновления для PointCast, да и компьютер, отягощенный программой, еле ноги волок – включая тогдашний чудо-Pentium 133. Поэтому миллионы пользователей, установив программу, практически тут же ее удаляли. Старая идея RSS лишь недавно смогла раскрыть свой потенциал.

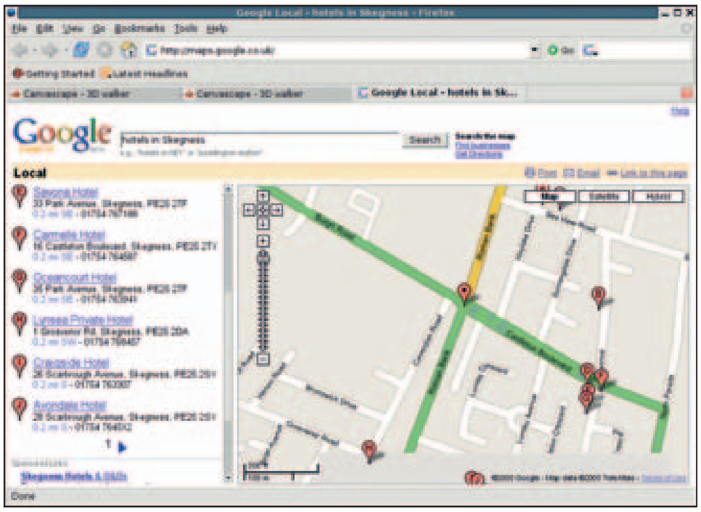
Новый порядок

О'Рейли недавно писал, что рекламные службы Google AdSense и Overture (ныне Yahoo Search Marketing) оттеснили DoubleClick, потому что они дотянулись «до самых до окраин» сети, а не только до ее центра. О'Рейли подразумевает, что сотня-другая гигантов сетевой публикации превращается в карлика на фоне явления, когда каждый пользователь становится полноценным участником сети: на фоне миллионов блоггеров, а также тех, кого The Register называет «педиафилами» [Прим. пер.: подробнее о «педиафилах» можно узнать на **http://en.wikipedia.org/wiki/User:Wiki-fiddler**].

Таким образом, фирмы Web 2.0 добиваются успеха в стиле «клиента мелкого, но много». Причем клиенты принимают активное участие в улучшении сервисов. Два напрашивающихся примера – Amazon и eBay.

eBay очевидным образом формируется пользователями; Amazon базируется на

>>



Google Maps – основной компонент популярных «сборищ».

ИНСТРУМЕНТЫ WEB 2.0 ДЛЯ НЕПОСВЯЩЕННЫХ

Толкование терминов: тэгирование, общественные закладки, синдикация, блоги и wikis

- **Тэггинг** (известен также как «фолксономия» (folksonomy), парафраз «таксономии» – хорошо бы эта кличка не прилипла) относится к процессу разбиения на различные категории фотографий в сети, записей блогов и другой информации, для облегчения процесса поиска. Например, фото утки, плавающей в пруду, можно описать следующими тэгами (ключевыми словами): 'утка', 'пруд', 'фауна' и 'парк'. После чего специальные поисковые сайты будут прочесывать блогосферу, чтобы найти опубликованные пользователями фотографии уток. [Прим. пер.: подробнее о фолксономии – на **http://en.wikipedia.org/wiki/Folksonomy**.]

- Delicious, Furl и другие сайты **общественных закладок** позволяют хранить в сети свои закладки, так что вы можете получить к ним доступ и управлять ими с любого компьютера, а также пользоваться ими совместно с другими пользователями. Если затем вы назначите тэги своим записям, то сможете облегчить поиск похожих закладок себе и другим пользователям. Например, можно узнать, какими еще страницами интересуются люди, посещающие ваш любимый сайт о волынках, или осуществить поиск по тэгу 'Uillean', чтобы разжиться информацией об ирландских трубках.

- **Синдикация**. RSS и Atom позволяют передавать по сети заголовки блогов и новостных сайтов как метаданные XML. Их можно читать, используя панели браузера, программы чтения RSS, различные web-сайты, или же просто включить эти данные в вашу домашнюю страничку.

RSS является центром персонализации сети Web 2.0. Он был взят на вооружение основными поставщиками новостей, типа BBC, лет пять назад. Поддержка RSS планируется в следующих выпусках *Microsoft Outlook* и *Internet Explorer*. Существуют две конкурирующие версии, по-разному расшифровывающие аббревиатуру RSS, и они вместе борются с Atom, разработанным Internet Engineering Taskforce (IETF). Большинство сайтов поддерживают RSS 1, но многие сайты предлагают все три формата.

- **Web-журналы** (web-logs, блоги) обязаны своему успеху использованием ссылок, которые образуют сообщества. Эти неформально-объединенные группы – идеальное место для поддержания контактов с единомышленниками и коллегами, а с помощью RSS вы можете увидеть их всех разом.

- Что касается **wiki**, достаточно сказать, что эти программы помогают пользователю запросто поучаствовать в создании сайта, значительно лучше, чем форумы и электронные доски сообщений. Неуклонное перемещение Wikipedia вверх по списку Top 100 – самое очевидное свидетельство ее успеха.

традиционной системе почтовых заказов, но предоставляет пользователям всевозможные шансы поучаствовать, от обзоров до ссылок на предпочтения людей, которые уже купили облюбованную вами книгу. Добавьте к этому выдачу по запросу адресов книготорговцев и букинистов – и вот вам сайт, просто рожденный для пользователей. Как может с таким конкурировать продавец из Web 1.0?

Концепция участия распространяется на все виды приложений – даже на системы распространения, вроде BitTorrent. О'Рейли утверждает, что *BitTorrent* «демонстрирует ключевой принцип Web 2.0: чем больше людей используют сервис, тем лучше он становится, причем автоматически». Этот принцип работает на новостных сайтах типа Digg, где новость публикуется лишь после голосования пользователей: дать ей «добро» или нет.

В разработке

Частично обвал дот-комов объясняется тем, что фирмы, получившие большие инвестиции, были заранее обречены – слишком много бурлило экспериментов, чтобы справиться с выбором надежной технологии и выстроить бизнес-планы. Похоже, что и в Web 2.0 происходят такие же эксперименты. Возьмем Technorati, сервис, осуществляющий поиск по блогосфере с использованием тэгов для обновления поисковой машины в реальном времени. Вместо приобретения популярности, Technorati погряз в технических проблемах. Поэтому большинство пользователей предпочитают его соперников, например, PubSub («опубликуй и подпишись»), который оповещает подписчиков в реальном времени, находя информацию, соответствующую запросу, в web-логах, новостных группах и других источниках. Подписка на PubSub бесплатна. Сервис живет за счет рекламы, венчурных капиталовложений и старого доброго оптимизма Web 2.0.

«WEB 2.0 – ТАКОЕ ЖЕ ПОЛЕ БИТВЫ, КАК И ПРОЧИЕ ОБЛАСТИ ПРИМЕНЕНИЯ ПО.»

Flickr: помещайте фото, описывайте их, делитесь ими… Революция!

Technorati не единственный известный в Web 2.0 сайт, испытывающий трудности. Похоже, что сервис по обмену и публикациям фотографий Flickr вообще не обновлял свое оборудование, даже после его прошлогодней покупки компанией Yahoo за 18 миллионов долларов. Поэтому сервис доступен урывками, как и некоторые домодельные сервисы Yahoo.

Похожие проблемы и у хостинга 6 Apart с его проприетарной блоггерской платформой начального уровня TypePad. Как-то в декабре он не работал почти целый день, а когда его починили, то обнаружилось исчезновение некоторых публикаций и комментариев.

Почти каждый сервис Web 2.0 – даже Flickr – находится в стадии бета, альфа или пре-альфа. Повсюду в сети сайты находятся в процессе разработки. Однако есть преимущества такого подхода. Microsoft требуется в среднем 3 года, чтобы обновить платформу – пользователи из-за этого разбегаются. А сайты Web 2.0 добавляют и удаляют предоставляемые возможности чуть ли не ежедневно, эволюционируя в соответствии со спросом. Такая модель разработки знакома пользователям Linux.

Закрытое или нет?

Internet всегда полагался на открытое программное обеспечение и открытые стандарты – TCP/IP, BIND, SMTP, Sendmail – и HTML, позволяющий каждому вступить в игру. Он остался свободным, потому что пропатченный HTTPd остается единственным выбором для web-сервера. Наверное, Web 2.0, развиваемый умниками и фирмами, выжившими после обвала дот-комов, пойдет по такому же пути?

Будем надеяться, что да, однако Web 2.0 представляет собой такое же поле битвы, как и прочие области применения программного обеспечения. На каждый *Flock* или Gmane, дарящих сообществу свои идеи и исходный код, найдется не открывающая код 37Signals, а также Google, скупающая патенты и сотрудничающая с AOL/Time Warner. Большинство Web 2.0 сайтов основаны на старом добром LAMP (*Linux*, *Apache*, *MySQL/PostgreSQL* и PHP/Python/Perl/Ruby – но список растет), и построены на применении открытого программного обеспечения прикладного уровня. Однако не все сайты предоставляют свободный доступ к своим сервисам. В перспективе, публикация GPL3 упрочила бы позиции открытого программного обеспечения, приравняв web-приложения к обычным приложениям с точки зрения лицензий. Но GPL3 предстоит еще долгий путь.

Открытые стандарты и программное обеспечение являются строительными блоками Web 2.0. Возьмем тэг <canvas> (холст). Холст – элемент HTML, создающий рабочую поверхность для любого вида графики (виджета, диаграммы, картинки и так далее). Функции рисования вызываются из JavaScript, поэтому изображения можно всячески преобразовывать (перемещать, масштабировать, поворачивать) в реальном времени с помощью простых скриптов. Подготовлено несколько JavaScript-библиотек, облегчающих создание всевозможных клиентских приложений, вплоть до весьма серьезных. Легче стало создавать упрощения страниц для тех пользователей, чей браузер не поддерживает тэг Canvas (*Explorer*, например).

«Линзы» Squidoo – тэгированная база знаний Web 2.0. Стиль несколько самодовольный.

Apple «придумала» тэг Canvas для виджетов Dashboard в OS X 10.4 (Tiger), и это позволило пририсовывать виджеты с помощью HTML и JavaScript. Некоторые web-разработчики, использующие Mac, сделали поддержку этого тэга в браузере *Safari*, чтобы встроить его в свои странички. Разработчики *Mozilla* тоже разглядели потенциал Canvas. Они его немного доработали, включили в состав *Mozilla* (и *Firefox*) и представили его организации Web Hypertext Application Technology Working Group (WHATWG), которая, в свою очередь, включила его в состав стандарта HTML 5.0, призванного облегчить разработку web-приложений. А потенциал у Canvas огромен – от представления гистограмм на лету до замещения Flash (нет нужды скачивать дополнительные модули) в несложных анимациях и играх.

Большой бизнес

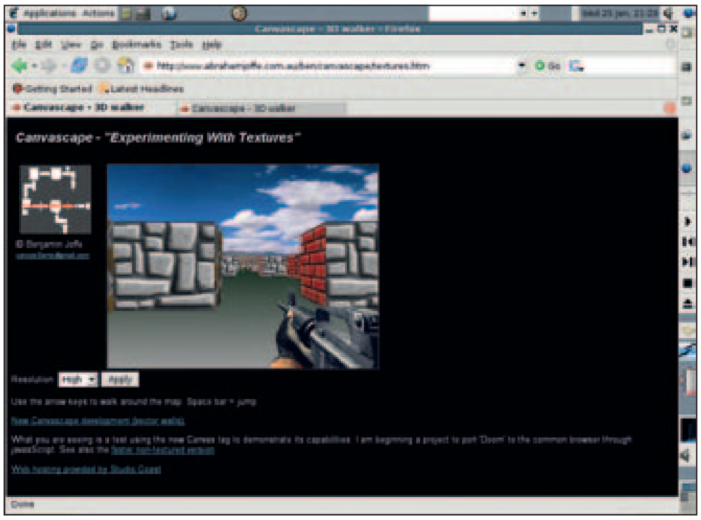
Когда Microsoft включит все возможности HTML 5.0 в новую версию своего браузера – неизвестно, однако фирма запоздало пытается догнать Ajax при помощи проекта 'Atlas' для платформы ASP.NET.

Введение новой технологии – лишь малая толика в сравнении с радикальной переменой курса Microsoft по отношению к Web 2.0. В ноябре прошлого года, через 10 лет после принятия решения не пренебрегать Интернетом, Microsoft анонсировала новую стратегию «сеть-как-платформа». Стратегия включает перезапуск инициативы Windows Over Web (начиная с Microsoft Office Live) на сайте (пока в бета-версии) **www.live.com**, API на основе SOAP для «сборищ» с применением MSN Search, и Virtual Earth (естественно, с ограничениями).

То, что Google (описанная О'Рейли как «оплот стандартов») заключила сделку с AOL, означает, что теперь у Microsoft действительно есть соперники – как и у eBay (Skype), и у Yahoo (Flickr, Delicious). И кто знает, какие еще сервисы захватят крупнейшие сетевые гиганты?

Для сетевых предпринимателей процедура IPO (первичная эмиссия акций) больше не является единственной стратегией выхода. Стратегия «сеть-как-платформа», использующая открытые протоколы и стандарты Интернета, позволяет снизить начальные вложения в проекты и дает больше шансов на быстрое расширение, чем медленно растущий рынок Linux-систем.

Это тот самый потенциал роста, свободный от обычных ограничений платформы, который притягивает венчурные капиталы. Вторая конференция, посвященная Web 2.0, прошла в октябре 2005 года и собрала 1.6 миллиона долларов на взносах участников, а число венчурных капиталистов превысило количество технарей. Шум, который наделали этакие деньжищи и две конференции Web 2.0, одних раздражает, других беспокоит. Эд Сим (Ed Sim) из Dawntreader Ventures советует предпринимателям смотреть сквозь рекламу: «Если вы предприниматель, перестаньте болтать о Web 2.0, подумайте лучше о том, как расширять свой бизнес и делать деньги».



3D-игра без установки дополнительных модулей! Оцените возможности тэга Canvas, блуждая в трехмерном лабиринте.

Открыто ли будущее?

Будет ли большая часть сети использовать свободное программное обеспечение, после того венчурные капиталисты назовут победителей и проигравших на пути к Web 3.0? Какая вам разница, свободен ли Интернет, если вы используете GNU/Linux, или Haiku, или NetBSD на своей рабочей станции?

Мы думаем, что разница есть. Независимый компьютер – это временная аномалия. Компьютерная сеть предоставляет несравнимые с ПК мощности. Большая часть программ, которыми вы пользуетесь каждый день, давно работает на удаленных компьютерах. (Главная причина затянувшейся доработки GPL3 – рассмотрение удаленно работающих программ, web- и других Интернет-сервисов.) Безразлично, работаем ли мы за полноценными компьютерами или простыми терминалами: если большинство приложений работают на удаленной машине, вопрос о разработке свободного программного обеспечения приобретает важное значение. Именно так будут работать приложения завтрашнего дня. А как же Web 2.0? Назовем его меткой на песке, указывающей путь к фантастической Семантической сети Бернерса-Ли (Berners-Lee).

Вернитесь к своему компьютеру и оглянитесь вокруг. Если у вас есть идея, воплотите ее. Создайте свой сайт. Используйте другие сайты для создания «сборища». Заведите дневник. Создайте ленту RSS и сообщите о нем миру. А теперь – бегом в Силиконовую Долину! Нацарапайте свои идеи на салфетке и помашите ею перед носом венчурных капиталистов. Вдруг ваш мыльный пузырь лопнет не сразу? LXF

НАЖМИТЕ ЗДЕСЬ

Исследуйте Web 2.0 прямо сейчас

- HTML тег **<canvas>** **www.whatwg.org/specs/web-apps/current-work/#canvas**
- Общественные закладки **Delicious** **http://del.icio.us**
- Общественные закладки **Diigo** с расширенными возможностями **www.diigo.com**
- Простой обмен фотографиями с помощью **Flickr** **www.flickr.com**
- Web 2.0 браузер **Flock** **www.flock.com**
- **FURL**. Еще одни общественные закладки **http://furl.net**
- **Gmane**. Двусторонний шлюз news-mail для рассылок разработчиков **http://gmane.org**
- **Mapki** Создайте свой гибрид на основе Google Map **www.mapki.com**
- **Piggy Bank** Превращает обычный браузер в семантический web-браузер **http://simile.mit.edu/piggy-bank**
- **Programmable Web** База знаний по различным сервисам, предоставляющих свои API для создания приложений-гибридов **http://www.programmableweb.com**
- **Open Laszlo** Открытая платформа для создания web-приложений **www.openlaszlo.org**
- **Проект Xanadu** Теда Нельсона (Ted Nelson) основан в 1960-х, это новая концепция гипертекста **http://xanadu.com**
- **Pub Sub** Основанный на подписке сервис, высылающий уведомления в реальном времени о найденных по запросу источниках **www.pubsub.com**
- **Simile** Семантическое взаимодействие метаданных и информации в средах unLike **http://simile.mit.edu**
- **Six Apart** Настоящий владелец LiveJournal **www.sixapart.com**
- **Technorati** Поиск по тэгам в блогах **www.technorati.com**
- **Writely** Обмен документами в сети **www.writely.com**
- **Zimbra** Пакет для совместной работы с интерфейсом AJAX **www.zimbra.com**
- **Zoho** Онлайновый текстовый процессор **www.zohowriter.com**

ДЛЯ ТЕБЯ

Как вы думаете, обоснован ли шум вокруг Web 2.0? Или нас ожидает еще один крах доткомов? Email: **letters@linuxformat.ru**?

Как выжать больше из ваших фотографий в Linux

Open Source предлагает все необходимые инструменты для создания и печати фотошедевров и обмена ими. Присоединяйтесь к **Полу Хадсону** и **Нику Вейчу**, которые научат вас этому искусству – не забудьте только сперва что-нибудь сфотографировать!

Компьютерщики и цифровые камеры неразделимы, как Ричард Столлмен и GPL. Широкие исследования *LXF* показывают, что большинство наших читателей после компьютеров более всего увлечены именно цифровой фотографией. Поэтому вне зависимости от того, фиксируете ли вы, как растет ваш ребенок, работаете приглашенным фотографом на свадьбе друга, или просто ведете сетевой фотодневник, скорее всего, фотоаппарат для вас очень важен.

Но, если честно, насколько полно вы используете Linux и Open Source для работы с фотографиями? Большинство дистрибутивов содержат все необходимые приложения для загрузки снимков в компьютер, их обработки и обмена с друзьями. Однако вам простительно знать эти приложения лишь поверхностно, поскольку они кажутся гораздо сложнее, чем есть на самом деле.

Но теперь перед вами полное руководство по цифровой фотографии в стиле Open Source. Мы расскажем, как подключить камеру к компьютеру, переписать снимки, исправить их и добавить спецэффекты, а также напечатать снимки и загрузить их в Интернет, чтобы удивить других. Все это работает в Linux и, разумеется, бесплатно.

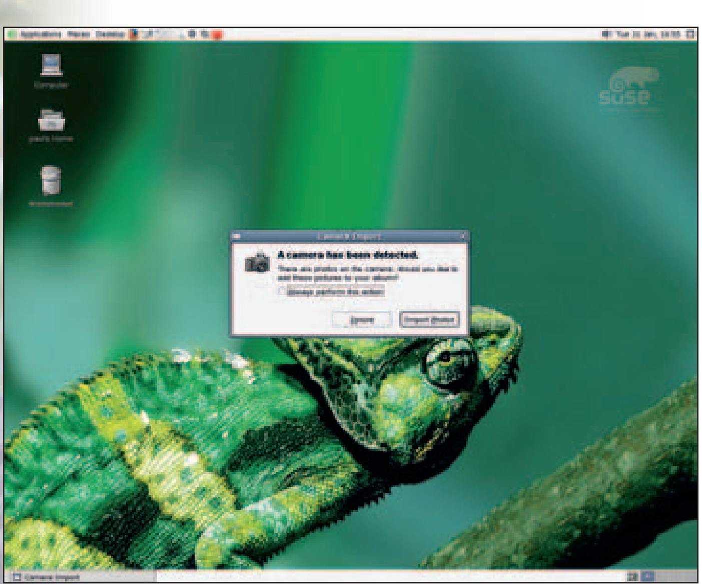
НА ДИСКЕ

На нашем диске вы найдете полный инструментарий для владельца цифрового фотоаппарата, включая все упомянутые здесь приложения, и даже кое-что еще.

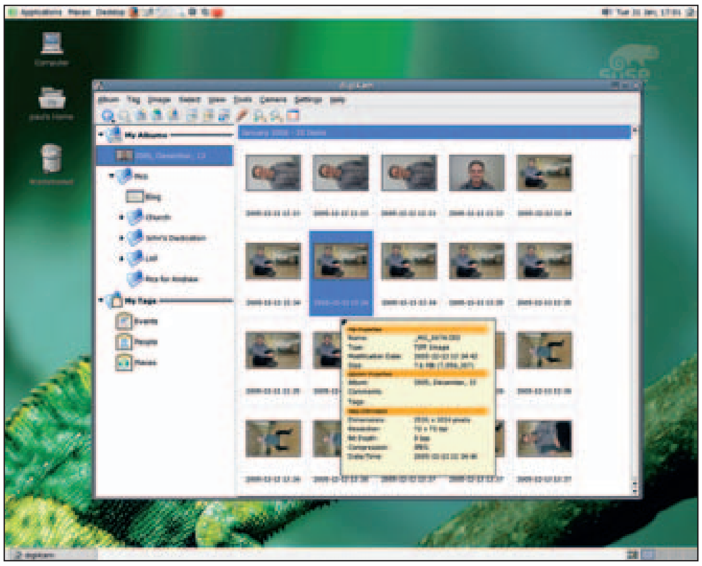
DCRaw
DigiKam 0.8.1
F-Spot 0.1.8
Gimp 2.2.10
GPhoto 2.1.6
Gtkam 0.1.13
Kalbum 0.85
KFlickr 0.5
Libgphoto 2.1.6
LittleCMS 1.15
QDVDAuthor 0.1.0
Scribus 1.2.4.1
UFRaw 0.6

Подключаем камеру

КАК ИМПОРТИРОВАТЬ СНИМКИ В LINUX



При подключении камеры Linux предложит импортировать снимки в ваш любимый менеджер фотографий.



***DigiKam* позволяет помечать снимки ключевыми словами и осуществлять по ним поиск, а также умеет импортировать фотографии с цифровой камеры.**

Поддержка цифровых камер в Linux всегда была на высоте, поэтому эта платформа великолепно подходит для постоянно растущего числа как фотолюбителей, так и профессионалов.

Несколько лет назад Linux воспринимал фотоаппараты так же, как и обычные USB-накопители – подключаешь камеру и списываешь файлы, вот и все.

Для исправления такого положения был создан проект Utopia, в рамках которого велась работа по интеграции оборудования с программной частью – так появился HAL (Hardware Abstraction Layer). Из-за того, что HAL первоначально был создан для GNOME, в наши дни при подключении цифровой камеры вы скорее всего увидите окно, как на рисунке сверху: Linux определяет камеру и предлагает импортировать снимки в фотоальбом. Приложения для управления фотоколлекцией отлично подходят для упорядочения и поиска снимков, среди них есть такие великолепные инструменты, как *DigiKam*, *F-Spot* и, конечно, творение Грэма Моррисона – *Kalbum*.

Решаем проблемы

Похоже, мы слишком торопимся. Что если при подключении камеры ничего не произошло? Этому могут быть три объяснения: **1)** Все работает, но компьютер не настроен на оповещение при подключении нового устройства; **2)** Ваш фотоаппарат настроен на неправильный режим соединения с компьютером; **3)** Наконец, ваш компьютер не настроен для работы с камерой.

Проверьте первый пункт, запустив *DigiKam* и открыв диалог «Добавить камеру» (Add Camera) в меню «Камера» (Camera). Нажмите кнопку автоопределения, и если все работает, то камера появится в списке устройств.

Многие модели камер имеют два режима передачи данных: соединение с ПК (также называется «нормальным» режимом) и PTP (иногда называется Print Mode – для печати напрямую с камеры), что означает Picture Transfer Protocol (Протокол передачи изображений). Особой разницы между ними нет, но иногда поддерживается только что-то одно. Таким образом, если ваш аппарат настроен на режим PTP и не определяется *DigiKam*, попробуйте переключить его в другой режим, или наоборот. Почитайте руководство к вашей модели камеры, хотя обычно нужный вам режим носит название, похожее на Communication Mode (Режим передачи данных).

Проще и быстрее всего перемещать изображения на ПК при помощи устройства чтения карт памяти (Card Reader)

Если у вас старый ПК

Наконец, проблема может лежать в вашем компьютере, который не настроен на работу с фотокамерами. Если ваш дистрибутив работает на старом ядре (например, ниже версии 2.6.10), то вам есть смысл обновить его, так как ядра версии 2.6.13 или 2.6.14 содержат ряд важных улучшений для домашних пользователей. Если вы не хотите обновляться, дайте от имени root команду *modprobe usb-storage*, и возможно вам удастся подмонтировать устройство; Вы также можете приобрести устройство чтения карт памяти за 20-30 долларов.

>>

ВЫБИРАЕМ СОФТ

Наш диск содержит достаточно программ для организации полноценной фотостудии. Если речь идет о фотоальбомах, обратите внимание на *DigiKam* и ее конкурента на основе GNOME/Mono – *F-Spot*. Для связи с фотокамерой оба приложения используют одну библиотеку – *Libgphoto2*, поэтому если одна программ у вас работает, со второй проблем тоже не будет. Помимо этого, для работы с RAW-файлами вам потребуется программа-конвертер DCRaw (*подробнее об этом на странице 60*). Для редактирования снимков лучшим выбором будет *Gimp*, поэтому мы советуем установить его и все дополнительные пакеты, включая опциональные расширения.

Для печати фотографий лучше всего установить настольную издательскую систему *Scribus* и дополнить ее системой цветокоррекции *Liblcms*.

<< Загружаем и исправляем фотографии

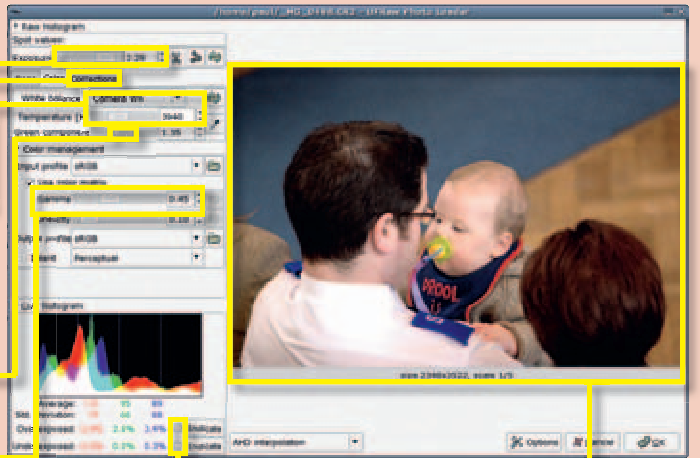
ПОДУМАЙТЕ ОБ ИСПОЛЬЗОВАНИИ RAW-ФАЙЛОВ

Скорее всего, ваша камера сохраняет снимки в форматах JPEG и TIFF, однако вы по достоинству оцените формат RAW, поддержка которого есть в профессиональных камерах, а также во многих полупрофессиональных моделях.

Формат RAW содержит информацию, напрямую снятую с матрицы вашей фотокамеры, которая потом обрабатывается на компьютере. Для сравнения: к JPEG-снимкам с самого начала применены баланс белого, настройка цвета, яркости и резкости, после чего все это еще и сжато. Поэтому, несмотря на то, что RAW-файлы громоздки и требуют ручной работы, они гораздо более гибки: вы можете менять выдержку, баланс белого, можете увеличивать резкость картинки с помощью продвинутых инструментов *Gimp* и многое другое. Вы также получите более широкий цветовой охват, так как многие RAW-файлы содержат 36-битный цвет вместо 24-битного в JPEG. Если ваша камера поддерживает формат RAW (и у вас имеется вместительная карта памяти), мы настоятельно рекомендуем вам его использовать.

В последнем случае вам понадобится специальный фильтр, который позволит импортировать снимки напрямую в *Gimp*. Выберите тот, который работает именно с вашей моделью камеры (большинство производителей используют собственные проприетарные версии RAW) – по нашему опыту, лучше всего работает фильтр UFRaw, который вы найдете на диске. Скомпилируйте его с помощью обычных **./configure**, **make**, **su** и **make-install** и вы будете готовы к работе. Запустите *Gimp* и попытайтесь загрузить RAW-файл. Если все работает, вы увидите окно *UFRaw*, где вам будет предложено конвертировать исходный файл. На этом этапе происходит главное волшебство: даже неудачные снимки можно «вытянуть» при осторожном использовании настроек.

НАСТРАИВАЕМ ЦВЕТА И ОСВЕЩЕНИЕ С ПОМОЩЬЮ UFRAW



Перетащите бегунок Exposure вправо, чтобы осветлить картинку.

Перейдите на вкладку Corrections, чтобы изменить насыщенность и контраст.

Баланс белого. Используйте значение Camera для сохранения исходных параметров, Auto для автоматического улучшения. Снизу есть бегунок для ручной настройки.

Многие камеры «уходят» в синие или зеленые тона при съемке. Исправьте это здесь.

Перетащите бегунок Gamma влево для подстройки яркости экрана.

Включите эти опции и *UFRaw* предупредит вас о потерях деталей при пере- или недоэкспонировании.

Смотрите изменения в реальном времени в этой части окна.

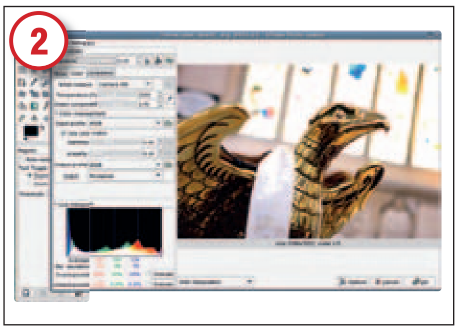
ИСПРАВЬТЕ ЭКСПОЗИЦИЮ

Существует два способа исправления недоэкспонирования в цифровом фото. Работа с форматом RAW позволяет сохранить детали, уменьшить цветовые перекосы, а также совершать все изменения за один прием, а не шаг за шагом. В *UFRaw* вы просто перетаскиваете бегунок экспозиции до тех пор, пока гистограмма не будет выглядеть гармонично.

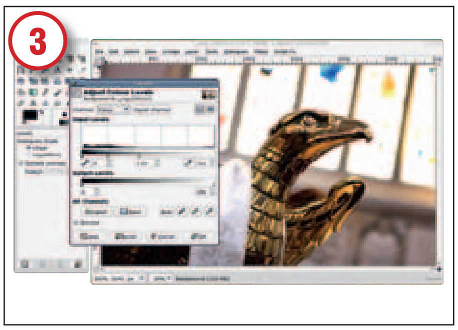
Если вам неоткуда взять фотографии в формате RAW, попробуйте поработать с 16-битными TIFF-файлами. Загрузите темную картинку в *Gimp* и настройте цветовые уровни (Инструменты > Инструменты цвета > Уровни). В окне «Уровни на входе» мы можем установить, какие цвета *Gimp* должен считать самыми светлыми, самыми темными и нейтрально-серыми (посередине).

Если ваш снимок, наоборот, засвечен, вы можете повторить те же процедуры, но с другими значениями: пользователи RAW передвигают бегунок выдержки вниз, все остальные меняют границу черного цвета в настройках уровней.

Камера правильно сфокусировалась на объекте, однако неверно измерила освещенность из-за бьющего с заднего плана света, так что объект получился недоэкспонированным. Гистограмма показывает, что почти все цвета находятся с левой стороны графика, что опять-таки говорит о недостаточной выдержке.



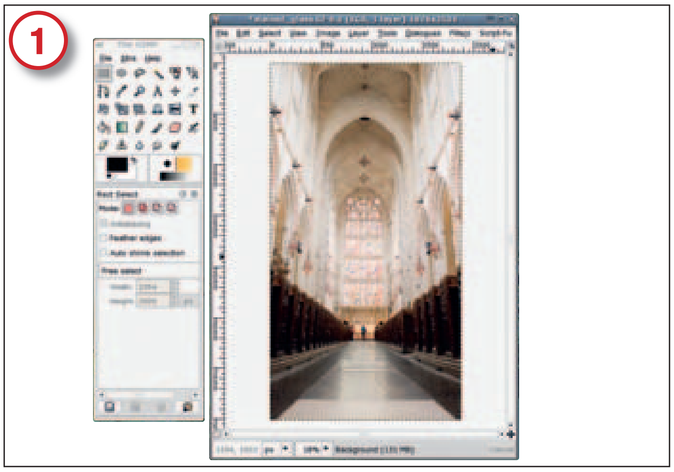
При редактировании RAW-файла, увеличение экспозиции высветляет задний план, однако в этом нет ничего страшного, так как он находится не в фокусе и выглядит приятно размытым (специалисты называют это «боке»).



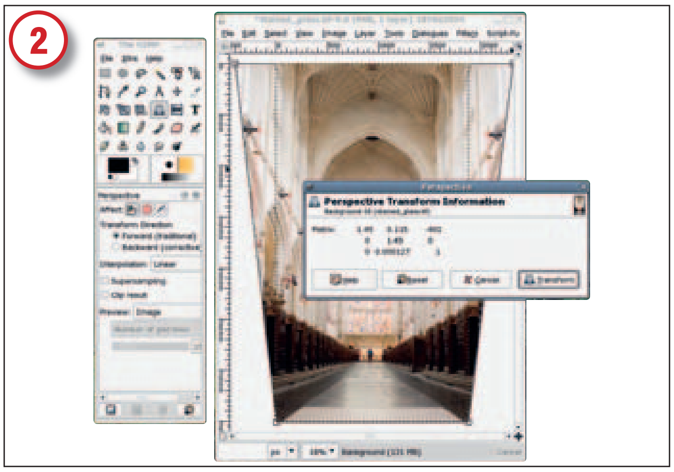
Не имея доступа к RAW-файлу, мы регулируем границы белой области в настройках уровней *Gimp*. Цвета, которые раньше были просто светлыми, теперь стали полностью белыми, так как изменение уровней «обрезало» часть цветового диапазона. Подобрав нужное значение для границы белой области, вы можете перетащить центральный бегунок, добавив тени там, где это необходимо.

ИЗБЕГАЙТЕ СХОДЯЩИХСЯ ВЕРТИКАЛЕЙ

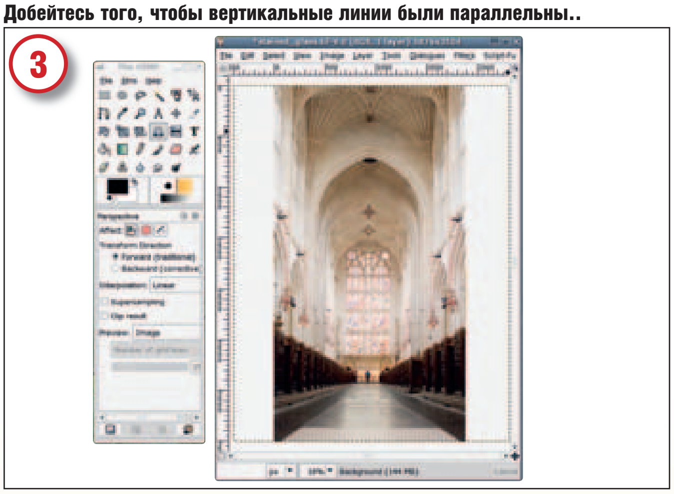
При съемке архитектуры вы обязательно столкнетесь с феноменом «сходящихся вертикалей». Если вы находитесь внутри большого зала и направляете камеру вверх, то на снимке верхняя часть зала будет меньше нижней. Можно отойти подальше, и тогда этот эффект будет не так заметен, однако существует два способа для исправления уже сделанных снимков. Вы можете купить специальный объектив с вогнутыми линзами за тысячу-другую долларов, либо бесплатно исправить этот недостаток в *Gimp*!



Мы сделали этот снимок, лежа на полу собора в Аббатстве Бат (Bath Abbey) и снимая снизу вверх. Обратите внимание, что потолок кажется уже пола, что является отличным примером сходящихся вертикалей.



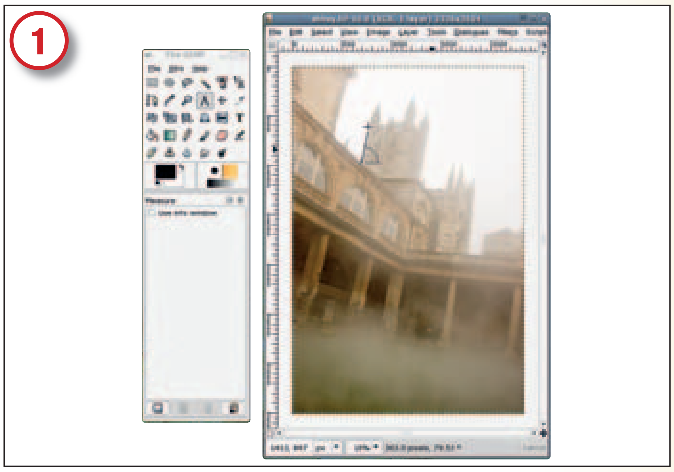
Перейдите в Инструменты > Инструменты преобразования > Перспектива. Углы изображения будут отмечены квадратами, и вы сможете изменять перспективу мышью в нужную сторону, компенсируя наклон фотографии. Добейтесь того, чтобы вертикальные линии были параллельны..



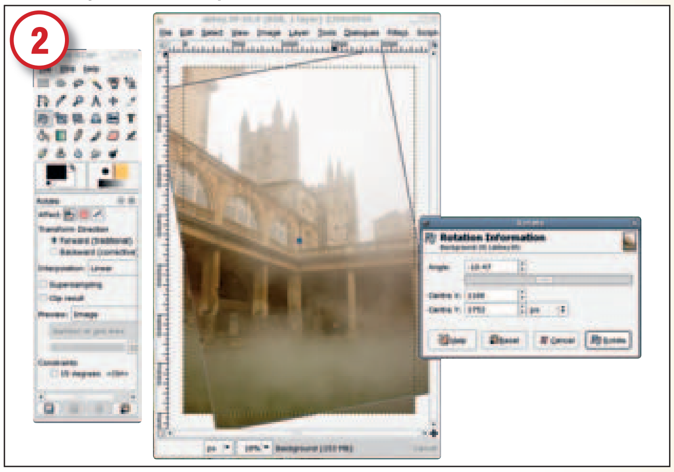
Выберите инструмент «Преобразование» и увеличьте фотографию для того, чтобы вернуть ей прямоугольную форму.

ИСПРАВЛЯЙТЕ УГЛЫ НАКЛОНА

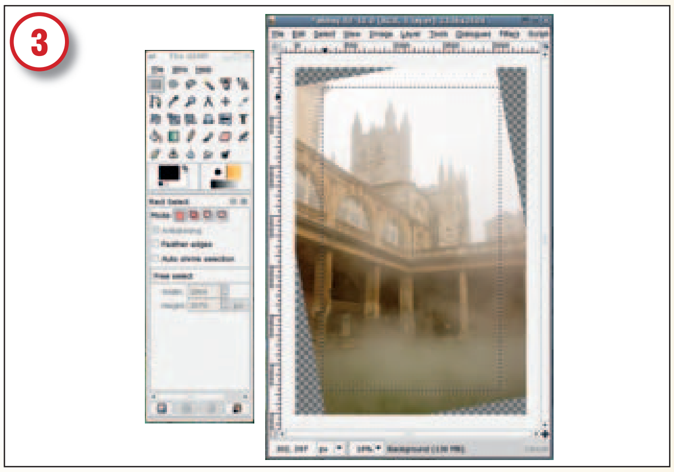
Съемка под углом способна придать динамику обычной фотографии. Придерживайтесь простого правила: угол надо рассчитывать как следует, в противном случае лучше снимать без наклона. Отклонение в 3 градуса выглядит как брак, тогда как 10-градусный и более наклон уже смотрится творчески. Тем не менее, иногда (слишком сильно наклонились, или ориентировались на другой объект) вам все же может захотеться повернуть картинку.



Выберите инструмент «Измерение расстояний» (значок в виде циркуля) и проведите линию вдоль объекта, который вы хотите выровнять. В строке информации появится угол, на который эта линия отклонена от горизонта (в нашем случае это 79,53⁰)



Для собственно поворота, зайдите в Слой > Преобразование > Параметры вращения. Теперь введите в поле нужный градус (у нас это 10,47⁰) и нажмите кнопку Повернуть.



После вращения по бокам картинки появились прозрачные области. Обрежьте их при помощи инструмента кадрирования.

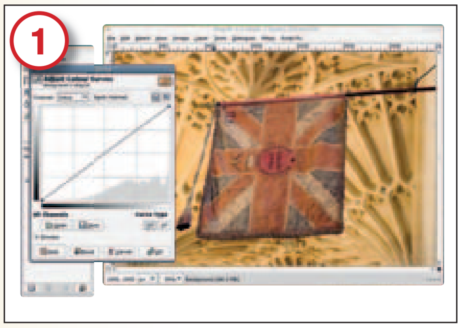
>>

Применяем спецэффекты

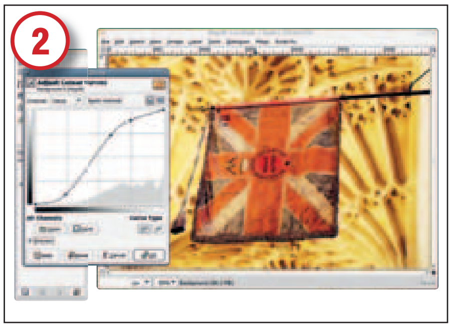
ДОБАВЬТЕ КОНТРАСТА

Поиск удачной композиции и выбор места съемки – это лишь полдела. Часто для того, чтобы изображение стало более выразительным, вам приходится прибегать к помощи *Gimp* для добавления специальных эффектов. Самый простой способ сделать иллюстрацию более выразительной заключается в значительном добавлении контраста, после чего цвета перестают быть реалистичными и начинают «цеплять» глаз. Более сложным эффектом является перекрестная проявка (используется при проявке пленки), когда негативы обрабатываются как позитивы и наоборот. В результате часть цветов инвертируется, что дает эффект соляризации.

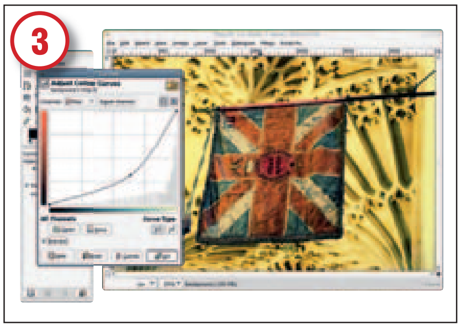
Чтобы увеличить контрастность, откройте окно настройки кривых (Инструменты > Инструменты цвета > Кривые). Вдоль горизонтальной оси располагается черно-белый градиент, показывающий текущее соотношение цветов. Градиент вдоль вертикальной оси отображает результат действия кривой. По умолчанию, градиент представлен равномерной полоской, где черное соответствует черному, белое – белому, а остальные цвета располагаются в образующемся пространстве. Для увеличения контраста нам нужно сделать темные цвета темнее, а светлые светлее.



Начните с щелчка мышью по месту, где первая горизонталь сетки пересекается с первой вертикалью. *Gimp* отобразит координаты точки в верхнем левом углу. Вы указываете на точку с координатами 63,63. Щелчок мышью создаст узловую точку, которую нужно перетащить вниз до позиции 63,20.



Теперь найдите пересечение третьих горизонтали и вертикали и потяните точку вверх до координат 192,220. Если у вас включен предварительный просмотр, вы заметите, что цвета стали гораздо выразительнее. Можно создать на кривой дополнительные точки и подстроить цвета более точно.

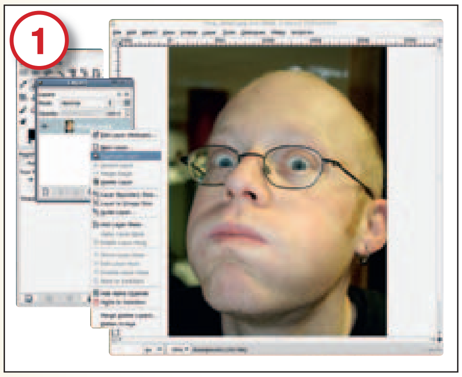


Чтобы добиться эффекта перекрестной проявки в *Gimp*, меняйте форму кривых для каждого из трех каналов RGB. Подсказка: лучше начать с контрастного изображения, поэтому есть смысл изменить кривые дважды.

НЕМНОГО РОМАНТИКИ

Кольца с эффектом размытия станут прекрасным дополнением к вашей оптике, позволяя выборочно размывать снимки и придавать им ощущение романтического крем-брюле. Однако они недешевы, и вы вряд ли купите больше одного, поэтому в ситуации, когда вам нужен мягкий фокус, а нужного фильтра нет под рукой, положитесь на *Gimp*.

А теперь возьмем фотографию, которая точно нуждается в романтическом настроении. Познакомьтесь с Элом Бикхемом (Al Bickham) – техническим редактором нашего родственного журнала – *PC Format*. Он не всегда так выглядит, кажется, сегодня он даже причесался... Как бы то ни было, с помощью небольшого волшебства *Gimp*, мы слегка размоем фотографию, которую Эл хочет поместить в интернет-форум знакомств.



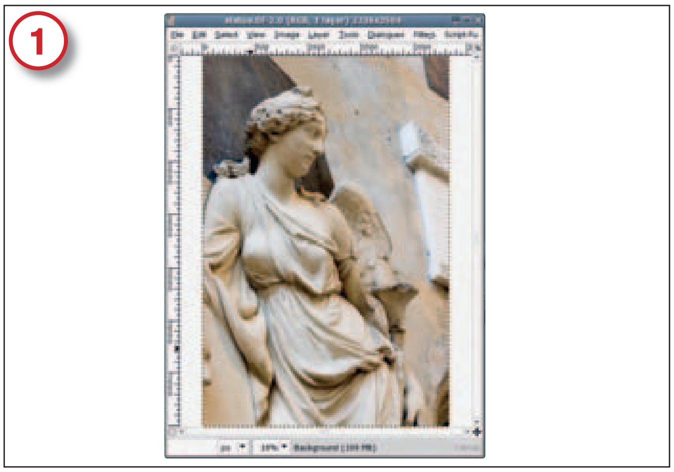
Сохраняйте копию исходного изображения, чтобы было потом с чем сравнить результат – размытые изображения лучше делать немного ярче оригинала, так нерезкость лучше смотрится при свечении. Нажмите Ctrl+L, чтобы вызвать на передний план панель слоев, после чего щелкнете правой кнопкой мыши по фону и выберите пункт «Создать копию слоя».

Убедившись, что у вас выделен дубликат исходного изображения, примените к нему фильтр Гауссово размытие (Фильтры > Размытие > Гауссово размытие), установив значение радиуса между 50 и 100. Чем выше это значение, тем более мягким будет фокус.

В меню слоев измените режим наложения верхнего слоя на Экранный (Screen). Теперь выделите нижний слой и настройте его уровни. Перетащите границу темной области так, чтобы общее изображение по яркости стало похоже на оригинал.

УМЕНЬШИТЕ НАСЫЩЕННОСТЬ ЦВЕТОВ

Цветность фотографии иногда отвлекает от основной идеи, которую хотел донести фотограф до зрителя – возможно, освещение в данном случае важнее конкретных цветов. Как бы то ни было, конвертация изображения в режим оттенков серого может дать снимку новую жизнь. Первый способ самый простой, но в *Gimp* у вас имеется дополнительный контроль над результатом.



На этой фотографии изображена мраморная статуя, которая сама по себе малоинтересна. Стена на заднем плане тоже из мрамора, поэтому статуя ничем особым не выделяется.

Чтобы получить обычную черно-белую фотографию, достаточно ее обесцветить. Это делается через меню Слой > Цвет > Обесцвечивание. Однако, как вы можете заметить, результат выглядит безжизненным.

Мы можем увеличить контраст и улучшить изображение путем редактирования кривых (Слой > Цвет > Кривые). Сместите нижнюю часть кривой ближе к черному (вниз), а верхнюю – ближе белому (вверх).

СМЕШАЙТЕ КАНАЛЫ

При черно-белой съемке многие фотографы используют цветные фильтры, чтобы регулировать поток света, проходящего через линзу камеры. Мы можем повторить этот эффект при помощи Channel Mixer в *Gimp*. Рассмотрим все возможности этого приема на примере фотографии Гаэля Дюваля (Gaël Duval) из *LXF70*, так как в ней одновременно присутствуют телесные тона (красные), листва (зеленые) и небо (синие).

Откройте микшер каналов, который располагается в меню Фильтры > Цвет и отметьте в нем пункты «Монохромный» и «Сохранять яркость». Для этой фотографии мы установили значения RGB как 200,0,0 для того, чтобы выделить лицо Гаэля.

Во второй раз мы установили значения RGB в 0,0,200 и получили более яркое небо, в то время как Гаэля стало трудней разглядеть из-за листвы. Такой метод позволяет исправлять цвета облачного неба.

Здесь мы перемешали цвета и отключили пункт «Сохранять яркость». Прибавка значений RGB в сумме составляет 130, поэтому фотография выглядит слегка засвеченной.

>>

<<

Обмениваемся фотографиями

ПОЛУЧИТЕ КРАСИВЫЕ ОТПЕЧАТКИ

Приятно сделать хороший кадр и улучшать его на компьютере до тех пор, пока он не станет совершенным. Но зачем все это нужно, если вы не можете показать его другим? Неважно, будет ли фотография маленькой и вытянутой под размер бумажника, или вставлена в рамку 20x30 см. В любом случае, самым популярным форматом снимков до сих пор остается отпечаток.

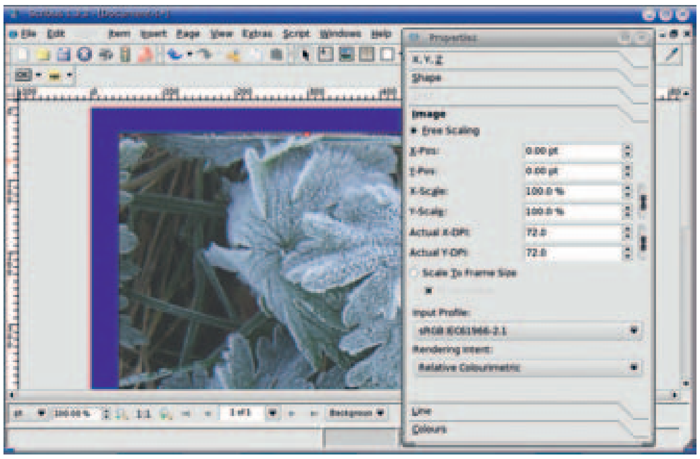
Большинство современных струйных принтеров способно печатать изображения в великолепном фотокачестве, но и здесь нужно помнить о некоторых подводных камнях, о которых мы здесь расскажем (*если вы хотите заставить ваш принтер работать в Linux, читайте статью «Железная работа вашего «железа» в LXF76, стр. 43*)

Качество изображения

Размер является важным параметром для достижения высокого качества печати. Понятно, что чем больше пикселей в картинке, тем лучше она будет выглядеть на распечатке. Изображение с низким разрешением может хорошо выглядеть на экране и ужасно – на бумаге, где будут заметны отдельные пикселы. В качестве совета (все зависит от оборудования, на котором вы печатаете) мы рекомендуем печатать картинки с разрешением от 200 до 300 ppi (пикселей на дюйм).

Снимок трехмегапиксельной камеры имеет размеры 2048x1536. Грубо говоря, этого достаточно для фотографии 20x30 см при печати на дому. Разрешения двухмегапиксельной камеры (1600x1200) хватит уже на меньший формат – не более 13x18 см. Конечно, никто вам не мешает распечатать трехмегапиксельный снимок в формате 13x18 см, но не думайте, что результат будет сильно лучше.

Некоторые люди путают пиксельное разрешение с количеством точек на дюйм (dpi) их принтеров. Многие принтеры способны печатать с разрешением 2400 dpi и выше, но это не то же самое, что печатать 2400 пикселей на дюйм. Каждый пиксел формируется множеством точек, которые придают ему нужный оттенок и яркость, поэтому точное число пикселей при печати зависит от алгоритмов смешивания чернил, типа бумаги и других параметров.



Встроенные профили можно извлечь из снимков, но вы также можете назначить их самостоятельно, выбрав конкретное устройство вывода.

Другое распространенное заблуждение относится к формату JPEG. Многие считают, что этот формат плох, так как в нем используется сжатие с потерями, удаляющее из картинки часть деталей. Это так, и последствия такого сжатия лучше всего видны на распечатке. Однако вы можете сохранить изображение в этом формате с минимальным сжатием, и тогда какого-либо ухудшения качества не будет. Файл при этом будет достаточно большим, но зато вы сохраните качество.

Цветокоррекция

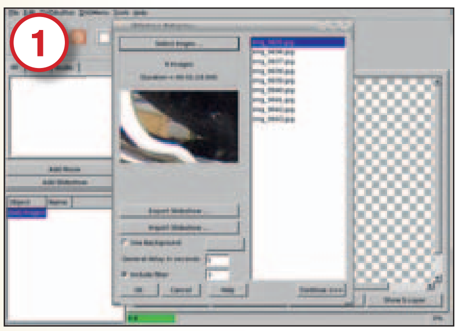
Для различных цифровых нужд – загрузки снимков в Интернет, создания слайд-шоу, презентаций и CD – снимки с камеры можно использовать без изменений. Печать, в отличие от вышеперечисленного, требует дополнительной коррекции изображений.

Изображение на экране формируется из точек трех цветов – красного, зеленого и синего, которые формируют изображения за счет света. Картинка на мониторе возникает за счет излучения, в то время как отпечаток на какой-либо поверхности сам отражает падающий на него свет. Это

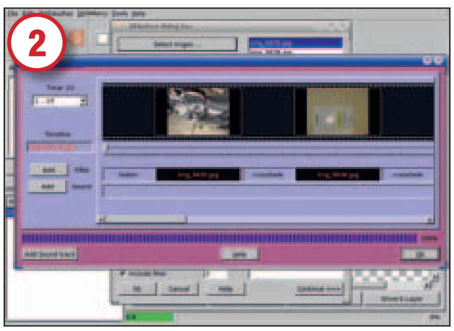
СОЗДАЙТЕ СЛАЙД-ШОУ НА DVD

Чтобы поделиться фотографиями с друзьями, не прибегая к распечатке, вы можете воспользоваться популярным средством – DVD слайд-шоу. Это современный аналог просмотра отпускных слайдов, когда у вас есть проектор и несколько скучающих гостей. Теперь все можно сделать без личного присутствия.

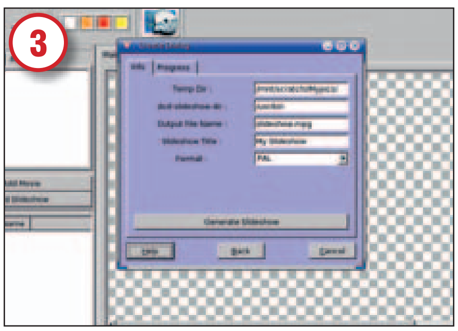
Вам понадобится программа с немудреным названием *DVD-slideshow*. Она предоставляет вам широкие возможности управления и при этом работает из командной строки. Если вам нужен графический интерфейс, обратите внимание на *QDVDAuthor*. Эта программа является бета-версией, но она работает неплохо, и вы можете найти ее на нашем диске.



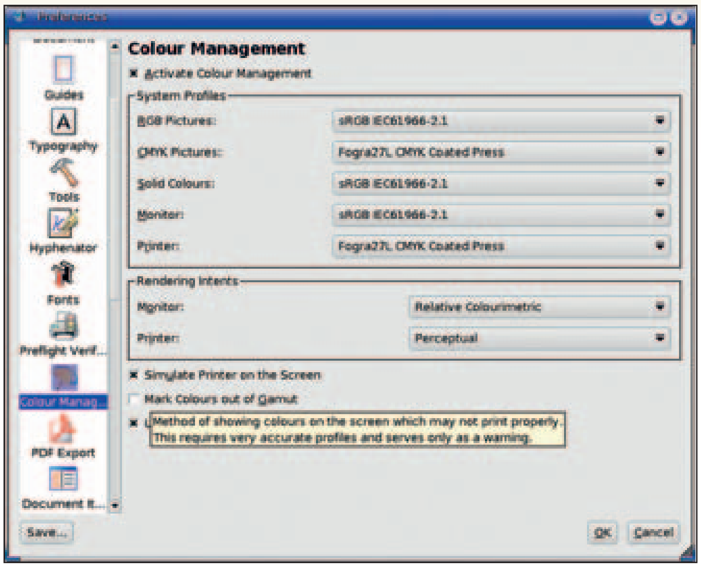
1 Запустите *QDVDAuthor*. Появится главное окно с несколькими панелями, по умолчанию отображающими новый DVD-проект. Здесь можно создать меню и добавить видеофрагменты, но мы ограничимся простым слайд-шоу. Из меню программы выберите пункт Add Slideshow Source (добавить источник изображений).



2 Убедитесь, что диск для временного хранения MPEG-анимаций для ваших фото не заполнен под завязку. Теперь нажмите кнопку Add Slideshow (Добавить слайд-шоу), добавьте изображения и задайте задержки между кадрами, а также эффекты при их смене.



3 Завершающее окно показывает, куда будет сохранен итоговый файл (имя и формат можно менять). Нажатием на кнопку Generate Slideshow (Создать слайд-шоу) вы создадите анимацию, которую можно записать на DVD-диск прямо здесь или в другой программе.



Активируйте систему управления цветом из панели настроек *Scribus*. Если такие настройки отсутствуют, придется пересобрать *Scribus* с включением *LittleCMS*.

очень важное отличие, так как видимое нами изображение состоит не из трех световых пучков, а из смеси чернил, которые поглощают одну часть падающего света и отражают другую. Поэтому цвета на бумаге формируются по принципу вычитания цвета.

В принтерах используются три основные краски – голубая, пурпурная и желтая, а также дополнительно черная краска для пропечатки темных областей.

Мы специально углубились в физику, чтобы объяснить, почему цвета на распечатанной фотографии могут не соответствовать вашим ожиданиям. Цифровые изображения зачастую содержат слишком много голубых и зеленых оттенков и при этом недостаточно контрастны. Вы можете вручную прибавить контраста в *Gimp* и настроить цвета соответствующими инструментами. Методом проб и ошибок вы подберете нужные значения, которые можно будет применять и в дальнейшем. Но это лишь самый приблизительный способ решить проблему с цветом.

В издательском деле в наши дни широко применяются цветовые профили, которые, в самых общих словах, описывают то, как следует интерпретировать числовые значения цветов. Чтобы эффективно воспользоваться этим, вам потребуются как сами профили, так и инструменты для работы с ними на всех этапах обработки снимка: отдельно для камеры, монитора и принтера. Последний этап затруднителен при использовании домашней техники. Цветовые профили как правило предоставляются изготовителями оборудования, но они совместимы только с фирменными драйверами, которых скорее всего нет для Linux. Однако управление цветом существует и для открытой платформы благодаря набору утилит *LittleCMS* (имеется на нашем диске и на сайте **www.littlecms.com**), которые даже встроены в *Scribus*.

Именно поэтому мы рекомендуем печатать из этой программы. Вам понадобится скопировать нужные цветовые профили в стандартную папку профилей *Scribus* – обычно это **/usr/lib/scribus/profiles**. После этого запустите программу и откройте секцию Color Management в окне Edit > Preferences. Здесь вы можете включить выбранные профили, которые будут действовать глобально в рамках приложения. Вместе с тем, у вас есть возможность применить профиль к конкретному изображению через его свойства в *Scribus*.

Документация к *Scribus* (**http://docs.scribus.net/index.php?lang=en&page=cms**) содержит массу полезных сведений об этом, включая то, откуда можно взять нужный профиль и как создать его самому.

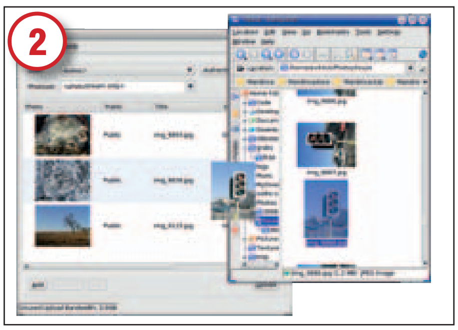
ОПУБЛИКУЙТЕ ФОТОГРАФИИ В ИНТЕРНЕТЕ

В мире появляется и исчезает много фотосайтов. Существовали сайты о фотопроявке и даже порталы производителей фотокамер. Но лучшим ресурсом для обмена снимками с друзьями и родственниками является Flickr.

Есть несколько поводов попробовать Flickr, даже если отвлечься от того, что эта online-служба работает под Linux. Во-первых, это бесплатно! Вы можете создать учетную запись и разместить на ней фотографии для всеобщего бесплатного обозрения. Единственным ограничением является объем файлов, который вы можете загрузить в течение месяца. Но зато в вашем распоряжении оказывается множество классных инструментов для манипулирования фотографиями, и вы можете общаться и обмениваться снимками с другими людьми.

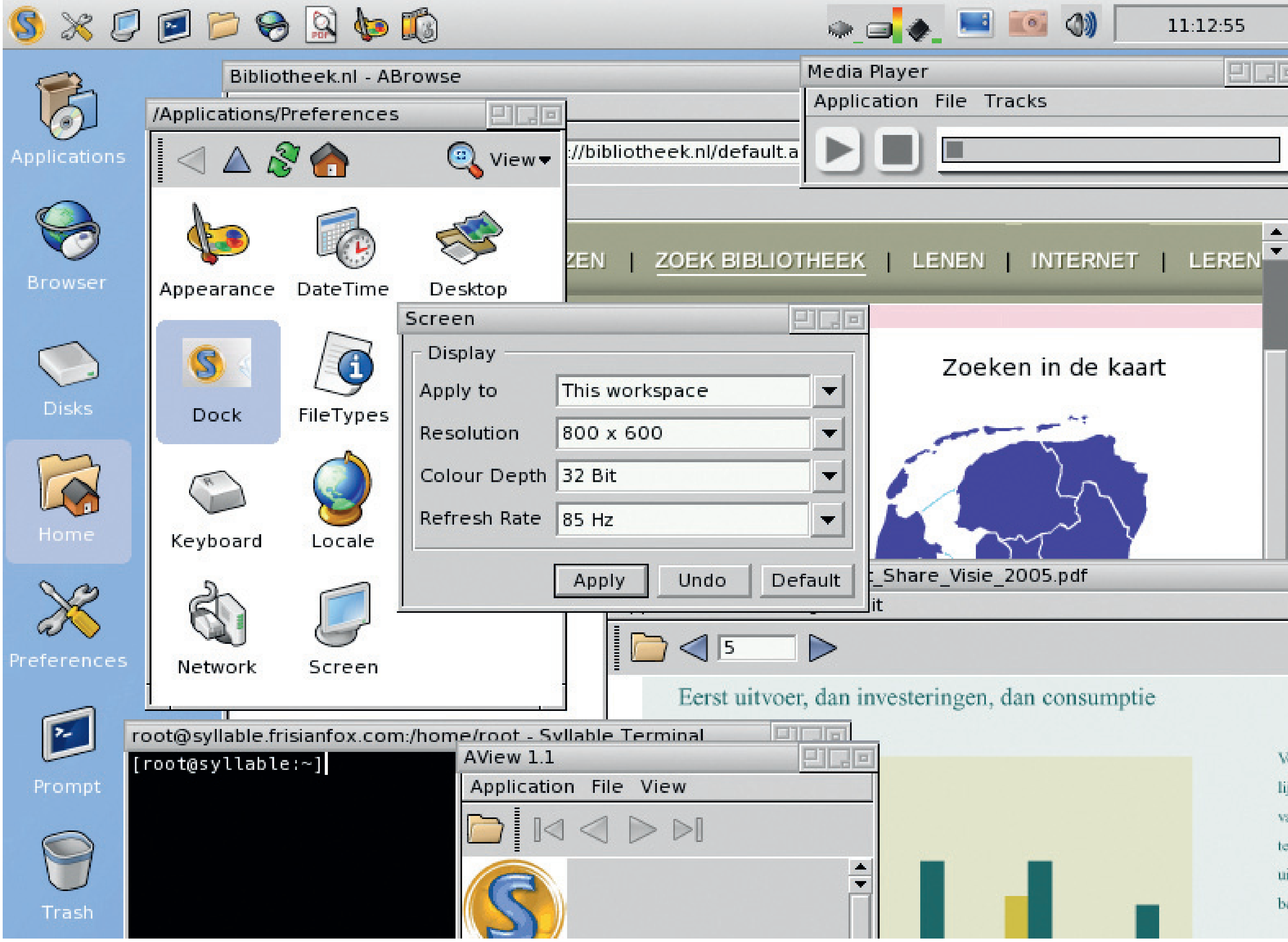
За скромную плату (менее 30 долларов) вы можете приобрести статус «Pro», по которому вам выделяется больше трафика и привилегий над простыми пользователями Flickr.

Присоединяйтесь к Flickr на сайте www.flickr.com. На данный момент для регистрации требуется учетная запись Yahoo. Вы можете приобрести учетную запись типа «Pro» в любой момент, а также пригласить друзей на просмотр ваших фотографий и даже сделать им отдельные учетные записи.



Вы можете загрузить снимки на Flickr как через сайт, так и с помощью отдельных инструментов. Наиболее удобной такой программой для Linux является *KFlickr*, которая позволяется добавлять к фотографиям теги (по ним можно искать снимки) и размещать снимки в общие папки, где они сразу же видны другим пользователям.

Можно разместить фотографии в разных группах или переслать их прямо на ваш блог (при его наличии). Теперь вы можете откинуться на спинку кресла и смотреть, как незнакомые люди изучают ваши фотографии и оставляют комментарии. Да, еще можно разрешить вашим друзьям делать то же самое.... LXF



Syllable: просто и со вкусом

НА ДИСКЕ

Молниеносная и простая в использовании операционная система, разработанная специально для рабочего стола – бывает ли такое? «Соучастник» Syllable Майк Сондерс представляет руководство по установке и использованию этого претендента на рабочий стол пользователей и на странице 59 рассказывает о человеке, стоящем за этой системой.

Вопрос на засыпку: назовите операционную систему, которая была бы быстра, свободна, стабильна и дружественна к пользователю. Ничего не приходит на ум, верно? Windows и Mac OS X привлекают простыми в использовании «мастерами» и инструментами настройки, но они и не свободны, и не так уж быстры. Linux, при всех сопутствующих ему замечательных технологиях, пока что недоступен пониманию «рядовых» пользователей, и изрядный вес его рабочего окружения и приложений не дает ему взлететь. Отличные системы BSD (Free, Net и Open), завоевавшие твёрдые позиции среди «компоманов», даже не пытаются прижиться на рабочих столах пользователей.

Стыд и позор: на наших глазах совместные усилия свободных разработчиков, охватившие весь мир, приводят к поразительным результатам благодаря мощи Open Source. Десятки тысяч хакеров работают над свободным ПО, постепенно захватывающим всевозможные рынки и дающим нам волю использовать и изменять программы. И за более чем 20 лет с тех пор, как Ричард Столмен изложил идею GNU – проекта, который привёл Linux к его сегодняшнему виду – продукт всё ещё не оспорил господство Microsoft и Apple на рынке пользовательских систем.

При всём изумительном мастерстве и объёме кода, разработанного в мире свободного ПО, наш лучший результат – непомерно сложная ОС, основанная по большей части на идеях 1970-х годов, слитая воедино группами, которые, в общем-то, вместе не работают, а зачастую даже и не общаются, и не используют стандарты, чтобы упростить свою же работу. Не поймите меня превратно: Linux – настоящий кладезь чудесного кода, но почему бы не стать ещё понятнее, проще и целостнее и не победить среди настольных ОС? И гораздо важнее: а если Linux это может, долго ли нам ещё ждать?

Посмотрите на Syllable – настольную операционную систему, изначально создававшуюся как дружелюбную к пользователю, быструю, аккуратную и свободную от излишней абстракции и устаревшей архи-

ГЛАВНЫЕ СВОЙСТВА

- Изначально задумана как настольная система
- Быстрое, компактное, SMP-совместимое ядро
- 64-битная журналируемая файловая система
- Программный интерфейс C++
- Современный объектно-ориентированный графический интерфейс и инструментарий
- Значительная совместимость с POSIX
- Набор инструментов GNU (*GCC*, Binutils и т.д.)
- Web-браузер, почтовый клиент, медиаплейер и многое другое
- Порты *Emacs*, Perl, *Apache*…

тектуры. Её цель – задействовать мощь разработки открытого кода без дробления; на практике это означает, что нет отдельных групп, занимающихся ядром, библиотеками, графическим интерфейсом, инструментарием и так далее. Вместо этого есть команда разработчиков, работающая над системой в целом, фокусируясь на ее логичности, а не на попытках принудить другие группы хакеров поддерживать то, что они делают. Мантра Syllable – «Думай о пользователях». Не о том, что *Glibc* хочет… а ядро не согласно… а *GTK* требуется… а разработчик *Pango* сказал… и так до бесконечности; просто о разумной ОС для конечных пользователей.

Из ниоткуда

Если вы уже глянули на экранные снимки, то могли подумать, что Syllable – Очередной Дистрибутив Linux. В конце концов, набор иконок Bluecurve широко используется в Red Hat, Fedora и других дистрибутивах Linux. Однако Syllable – независимая ОС: у неё своё ядро, библиотеки, графический интерфейс пользователя, инструментарий и т.д., всё разрабатывается синхронно, так что важные изменения затрагивают всю систему в целом. Тем не менее, команда Syllable (**www.syllable.org**) не тратит силы зря: они используют зрелый компилятор GCC для сборки и проектируют ядро так, чтобы проще было портировать драйверы Linux.

Syllable как таковая – отпрыск другого проекта: ныне покойного AtheOS. Норвежский программист Курт Скауен (Kurt Skauen) представил AtheOS миру как свой проект открытой операционной системы в марте 2000 года, после нескольких лет самостоятельной работы над ней. Прошло немного времени, и сторонние хакеры начали присылать ему свой код.

Однако Скауен лишь немногое брал от сообщества – его право, конечно – так что многие из взявшихся поддержать AtheOS переключились на разработку вспомогательных программ. В итоге, уникальная система, вобравшая концепции AmigaOS и BeOS, в значительной мере осталась работой одного программиста.

Возрождение ОС

В конце концов этот «театр одного актёра» привёл AtheOS к гибели. Тотальный контроль Скауена над системой и его приверженность к другим занятиям, вроде самолетовождения, приводили к длительным – часто многомесячным – периодам застоя. Другие разработчики чувствовали свое бессилие помочь развитию этой перспективной ОС, и в конце концов ситуация дошла до точки кипения. В июле 2002 года, после долгого молчания Скауена, Кристиан ван дер Влиет (Kristian Van Der Vliet) и несколько других разработчиков объявили о ветвлении или «возобновлении» AtheOS, – получив исходный код, распространяющийся по лицензии GPL и названный Syllable, они надеялись на значительное ускорение прогресса.

ПОЗНАКОМЬТЕСЬ С КОМАНДОЙ

- **Кристиан ван дер Влиет** (Kristian Van Der Vliet), он же Вандерс (Vanders). Лидер проекта. Проживая в Англии, Вандерс отвечает за общее направление развития Syllable и выпуск релизов. Работает над портированием *Glibc*, ядром и многим другим.
- **Арно Кленке** (Arno Klenke). Плодовитый немецкий программист, его работа – куча основных возможностей Syllable, а именно *ABrowse*, рабочий стол и многие драйверы.
- **Кай де Вос** (Kaj de Vos). Голландский разработчик системы сборки программ – *Builder* (родня систем Ports и *Portage*).
- **Хенрик Исакссон** (Henrik Isaksson). Шведский хакер, работающий над appserver – слоем графического интерфейса в Syllable.
- **Рик Каудилл** (Rick Caudill). Занимается инструментами рабочего стола и редакторами исходного кода.
- **Брент П. Ньюхолл** (Brent P. Newhall). Дизайн и разработка **www.syllable.org** и вспомогательных сайтов.

Разработчики Syllable в Лондоне на SylCon, февраль 2005 г., самая первая встреча.

Последствия сказались немедленно. Сразу же нашлись активные разработчики, обновляющие и сопровождающие код, и эта предельно компактная настольная операционная система получила шанс засиять. Ван дер Влиет – больше известный как Вандерс (Vanders) – и его команда переместили акценты AtheOS с любительской разработки на ОС, доступную для каждого конечного пользователя.

Несколько следующих лет команда Syllable исправляла, обогащала, добавляла и обновляла функции во всей базе кода, пока не достигла того, что мы видим сегодня: удивительно быструю и дружественную настольную операционную систему, с огромным потенциалом.

Вы можете попробовать Syllable 0.6.0a прямо сейчас, установив её с нашего диска. Номер версии указывает, что она ещё на стадии разработки, поэтому неизбежны ошибки и нехватка возможностей. Мы предлагаем ОС в двух вариантах, предоставляя вам возможность исследовать её либо на реальном оборудовании, либо на эмуляторе, например, *VMware Player*. Разобравшись с установкой, поговорим о том, как части системы сливаются воедино, что вы можете сделать, чтобы присоединиться к проекту, и что ждёт его впереди. Выражаясь бессмертными словами Гагарина – поехали!

>>

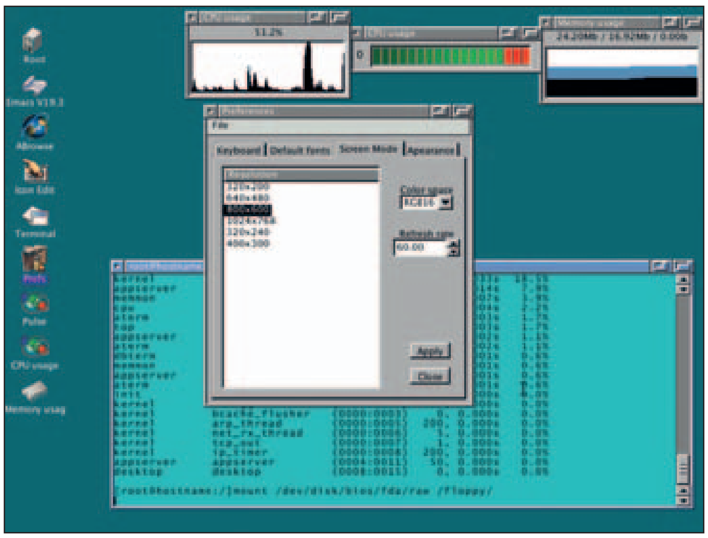
БЛАЖЕНСТВО УСТАНОВКИ

Результаты строгого курса на простоту использования

Рассмотрим на два типичных момента:

- **Установка программ**. Способ размещения программ в репозиториях, используемый в Linux, просто даёт определённые версии приложений в определённое время, кроме случаев, когда вы рискнули перейти на ветвь для разработчиков. Прозрачность Syllable можно сравнить с запуском музыки или фильма на CD – распакуйте программу в папку Applications, и она готова к работе. Никаких нудных библиотек, норовящих «рухнуть» на новом релизе, никаких разбросанных по всему диску файлов, никаких мучительных ожиданий, пока разработчик вашего дистрибутива соберёт нужные пакеты.
- **Установка оборудования**. Трудности с Linux вызваны бесконечной путаницей из-за отличий в версиях программ, ядер, слоёв файловых систем, стартовых сценариях и так далее. В Syllable заставить работать новое оборудование предельно просто – сбросьте файл драйвера в папку и перезагрузитесь. Не требуются ни «кривые» оболочки для двоичных драйверов, ни заголовочные файлы ядра, и при обновлении системы её работа не нарушится: Syllable заботится о двоичной совместимости. Может, это и не по душе борцам за чистоту свободного ПО, но так значительно легче жить.

Это лишь два примера. Поразительная скорость загрузки Syllable (менее 10 секунд на системе с 1.5 ГГц-процессором), единственный рабочий стол и набор графических инструментов, графические утилиты настройки – все эти факторы обуславливают создание ОС, отлично оборудованной для рабочего стола.

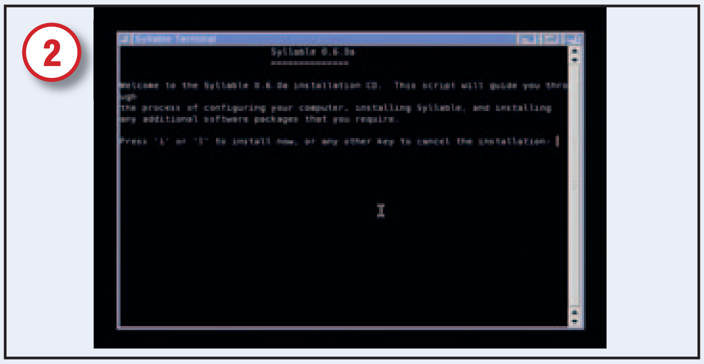


Образец торжества компьютерных наук, AtheOS Курта Скауена, с точки зрения интерфейса смахивала на ободранную автобусную остановку…

Устанавливаем Syllable

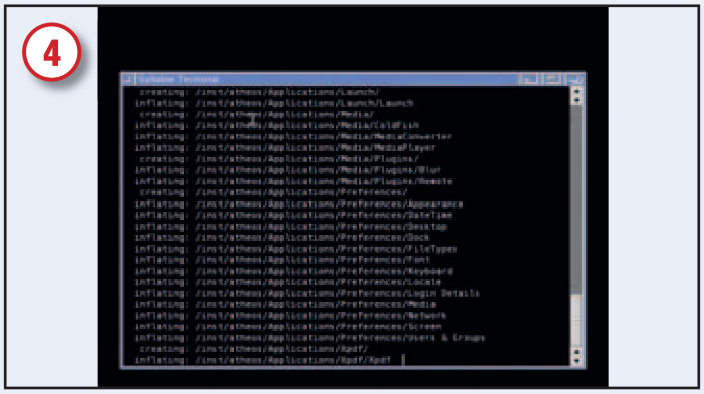
Syllable устанавливается на ПК аналогично любой другой ОС, то есть для неё требуется собственное место на диске. Прежде чем заново разбивать свой диск, попробуйте эмулятор *VMware*. В разделе «System» нашего диска находится *VMware Player*. Выполните установку, как описано на стр. 106. Распакуйте **syllable-0.6.0a.zip** (в разделе «Distros/Syllable»), запустите *VMware Player* и откройте **Syllable.vmx**. Syllable загрузит образ виртуального жесткого диска с предустановленной ОС, и вы можете прекратить читать эту страницу.

Для непосредственной установки Syllable на ваш ПК загрузитесь с CD или, если вы взяли наш DVD-диск, запишите образ **0.6.0a-basic.iso** на CD-R и загрузитесь с него. В идеале, Syllable требует процессора не ниже 233 МГц с 64 МБ памяти и VESA-совместимую видеокарту – но установка не будет работать с USB-клавиатурой. На показанном экране нажмите <Enter>, и начнется инсталляция.

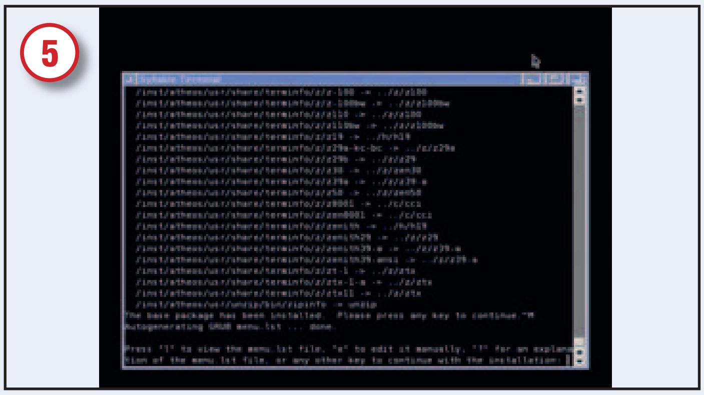


Помелькают сообщения о загрузке (см. «Часто задаваемые вопросы» на стр. 57, если загрузиться не удастся), и вы попадёте на этот экран. Инсталлятор Syllable в настоящее время – простой инструмент, основанный на скриптах: пока он ориентирован на разработчиков, но будет полностью переделан к релизу 1.0. Фанаты Amiga заметят знакомые рамки окон! Для продолжения установки, введите <I> (заглавная i).

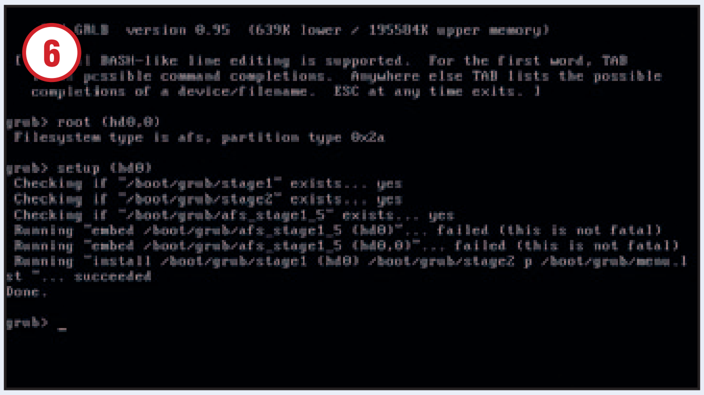
Syllable предложит создать новый раздел – нажмите Y. Пора выбирать диск (обычно hda или hdb) и раздел для установки системы. Установите тип раздела как «AtheOS», а его размер – не менее 500 МБ. Поставьте «птичку» слева, щёлкните «OK», затем «Quit», чтобы выйти из редактора.



Сценарий установки попросит подтвердить выбор раздела для установки: нажмите Enter, если предложенное значение вас устраивает, или введите имя другого раздела. Полный путь к разделу будет таким же, как в редакторе разделов на предыдущем шаге. Теперь инсталлятор приступит к копированию файлов ОС на ваш диск. На это потребуется несколько минут.



После установки вам будет предложено настроить файл конфигурации загрузчика. Syllable использует *Grub*, вероятно, уже знакомый вам. Нажмите E, чтобы открыть текстовый редактор и подтвердить загрузку с ранее сконфигурированного раздела; должны подойти установки по умолчанию. Выйдите из редактора, нажмите Enter и подготовьтесь к перезагрузке.



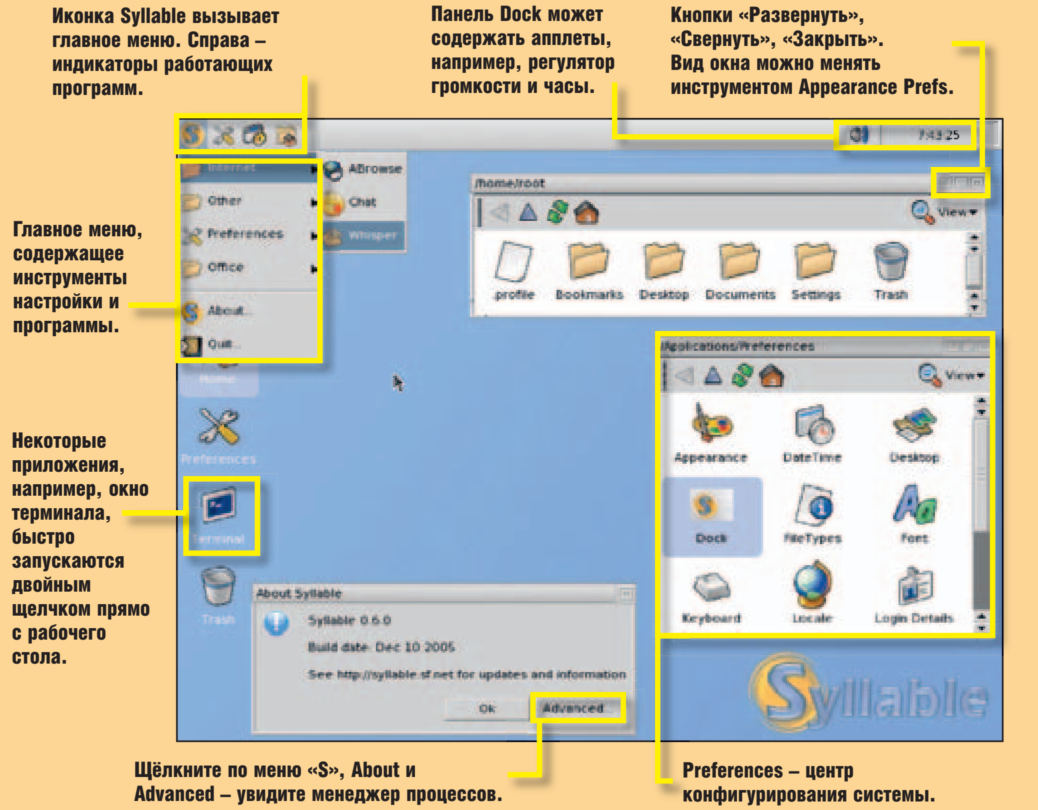
Снова загрузитесь с CD/DVD. Когда появится *Grub*, нажмите любую клавишу, чтобы остановить отсчёт времени до старта загрузки, затем – C, чтобы войти в командную строку. Введите **root (hd0,0)** (или **hd0,1**, или **hd1**, смотря куда установлена Syllable), затем **setup (hd0)**, чтобы установить *Grub*. Теперь можете извлечь CD и загрузить Syllable с жёсткого диска.

Первые шаги

При первой загрузке вам, возможно, придётся подбирать с пола свою отвалившуюся челюсть – Syllable потрясающе быстра. На любом ПК, изготовленном в последние несколько лет, можно ожидать загрузки менее чем за 10 секунд, с момента нажатия <Enter> в меню Grub до графического экрана, запрашивающего пароль. Проверка оборудования, загрузка драйверов, настройка сети и USB, запуск звуковой и графической подсистем – с нуля до работоспособного состояния – выполняется за ничтожную часть от времени, требующегося Windows и большинству дистрибутивов Linux. Секретов никаких: просто грамотное проектирование и аккуратный порядок выполнения стартовых сценариев.

Для входа в систему пароль пользователя root – «root», пароль guest – «guest». Повторяем: пароль для root – «root», для guest – «guest». На данном этапе разработки можно легко войти в систему под именем root и получить полный доступ к системе; в последующих релизах планируется обеспечить более высокий уровень безопасности.

ИЗУЧАЕМ РАБОЧИЙ СТОЛ SYLLABLE



Что включено

Поскольку Syllable нацелена на рынок SOHO (малый и домашний офис), в ней предусмотрен набор офисных приложений и утилит: web-браузер, почтовый клиент и т.д. Подобно самой ОС, все они находятся на различных стадиях разработки, хотя многие уже пригодны для основных задач.

- **ABrowse**. Web-браузер Syllable основан на *KHTML*, движке, который используется в браузерах *Konqueror* (KDE) и *Safari* (Apple). Мгновенно запускаясь, этот браузер соответствует по скорости рабочему столу Syllable, однако ему не помешала бы возможность индивидуальной настройки и дополнительные функции.
- **Whisper**. Клиент электронной почты, в настоящее время перерабатываемый, находится на стадии альфа-версии, однако некоторые разработчики Syllable уже сейчас используют его для отправки сообщений в списки рассылки. *Whisper* обеспечивает поддержку несколько учётных записей POP3, встраиваемых (inline) вложений, а также импорт из *Microsoft Outlook* (как и из *Express*). В финальном релизе появятся фильтры.
- **ColdFish**. Основанный на плей-листах музыкальный проигрыватель. Он похож на *XMMS* и *WinAmp* (однако попроще), и снабжён рядом модных средств визуализации, чтобы добавлять крутые эффекты к вашим песням.
- **Media Player**. Общий инструмент для воспроизведения видео- и аудио-файлов различных форматов, на базе библиотеки *FFMPEG*. Поддерживаются AVI, MPG, MP3 и другие форматы.
- **Утилиты**. В Syllable присутствуют текстовый редактор *AEdit* и калькулятор *Albert*. Ближе к релизу 1.0 эти имена заменятся в меню более описательными (например, «Image viewer», а не «AView»).
- **Настройки**. 13 небольших инструментов конфигурации почти полностью исключают потребность в командной строке. Используйте их для настройки внешнего вида рабочего стола, размера экрана, системных учётных записей и прочего.

>>

ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ

В Почему Syllable не загружается?

О Если из загрузочных сообщений видно, что Syllable зависает при обращении к жёстким дискам, попробуйте изменить параметры загрузки: в меню *Grub* нажмите E, чтобы приостановить отсчёт времени, опять E – для доступа к параметрам ядра, выделите строку, соответствующую ядру, и нажмите E снова. Добавьте в конец строки **enable_ata_dma=false**, затем нажмите Enter+B для продолжения загрузки. Если не удаётся загрузить графический интерфейс (appserver), сделайте, как описано выше, но добавьте к строке ядра **disable_pci=true**. Вы можете добавить эти параметры на постоянной основе, отредактировав после установки **/boot/boot/grub/menu.lst**.

В Почему нет звука?

О Если вы ничего не слышите, откройте Preferences > Media и выберите вашу звуковую карту из списка. Щёлкните на «Default Audio Output», затем – на кнопку «Controls», чтобы настроить микшер.

В Как я могу «убить» программу?

О Если программа зависла и не реагирует на кнопки нормального завершения работы, щёлкните по кнопке меню «S», About, Advanced, и попадёте в Syllable Manager. Перейдите на вкладку Processes – это список запущенных программ, которые вы можете остановить, используя кнопку «End Process» внизу.

В Где я могу найти инструменты разработки?

О Будучи настольной системой, Syllable не включает по умолчанию инструментарий GNU; вы можете найти его в пакете Developer Delight на нашем диске. Он включает *GCC*, G++, *Make* и многое другое. Необходимые для разработки «родных» приложений заголовочные файлы Syllable включены в базовую систему. См. каталог **devel** нашего диска, там есть дополнительная информация.

Движемся дальше

Syllable, возможно, ещё молода, но в отличие от ряда любительских проектов, она не ограничивается единственным козырем вроде текстового редактора или небольшой утилиты. Программисты, как пчелки, трудятся над обеспечением для Syllable широкой программной базы, включая такие пользовательские приложения, как средства работы в Интернете и аудиоплейеры, и благодаря их стараниям установка приложения выполняется очень просто. Программы Syllable делятся на две группы: «родные» приложения, разработанные специально для Syllable, и порты – существующие программы (обычно, инструменты командной строки Unix/Linux), они требуют различных способов установки. Родные приложения (в формате zip) можно просто распаковать в каталог **/Applications**, и они попадут в меню Syllable. Пока что такие приложения найти сложновато, но ситуация меняется в лучшую сторону – см. раздел «Ресурсы» в конце этой статьи, где даны ссылки на сайты с ПО.

Портированные программы обычно поставляются в формате tar.gz или zip, причём для последних релизов предпочтителен zip. Syllable применяет грамотную систему символьных ссылок, создающую уверенность, что программы в стиле Unix хорошо интегрируются в файловую систему, вместо попыток «подогнать» их под собственный дизайн Syllable с её **/Applications, /Documentation** и т.д. Если у вас есть портированное приложение в формате tar.gz или tgz, откройте окно терминала и распакуйте его в **/usr**:

```
cd /usr
tar xfvz /путь/к_файлу/filename.tgz
pkgmanager -a <созданный_каталог>
```

Эти команды распакуют программу в отдельный каталог в **/usr**. Последняя строка особенно важна: команда *pkgmanager* регистрирует вновь созданный каталог программы, устанавливая необходимые связи в файловой системе, чтобы другие Unix-программы могли с ним взаимодействовать. Чтобы удалить приложение, введите **pkgmanager -r directory**, затем удалите каталог.

Если ваш порт в формате zip, просто введите:

```
installport filename.zip
```

Эта команда автоматизирует описанный выше процесс, так что вы можете запускать программу непосредственно из **/usr**. Поэкспериментируйте с некоторыми из портов с нашего диска: мы поместили SDL-приложения (портированные автором), включая *DOSBox* и *SuperTux*. Многие порты поддерживаются в системе, называемой *Builder*, во многом подобной системам *Portage* из Gentoo и Ports из BSD. *Builder* автоматизирует процесс распаковки, наложения заплаток, компиляции и установки программ, и теперь способен собирать большую часть самой операционной системы.

Будущее

Итак, куда движется Syllable? Текущие планы развития охватывают этапы разработки, начиная с выпуска релизов серии 0.6.x. Поддержка печати, одна из основных функций, готова для внедрения в ближайшие несколько месяцев – Вандерс работает над тем, чтобы интегрировать CUPS в систему простым и понятным способом. Подсистема безопасности SylSec тоже должна быть закончена и включена в данную серию, наряду с графической библиотекой *lib2d*.

Релизы 0.7 и 0.8 отшлифуют базовую систему и включаемые в поставку программы, а также позаботятся о локализации и интернационализации. Имея стабильный API, разработчики приложений смогут создавать свои программы в готовом к выпуску состоянии: отсутствие фундаментальных изменений в API означает гарантию совместимости. В серии 0.9 не ожидается появления новых функций, но прежде чем 1.0 будет готов увидеть свет, выйдет серия бета-релизов, исправляющих ошибки.

Abrowse **– браузер Syllable на базе** ***KHTML*****. Внизу –** ***ColdFish*****.**

При нынешней скорости разработки Syllable 1.0 должна появиться через пару лет – но если к проекту присоединятся новые программисты, это произойдет раньше. На пару лет запланирована поддержка 1.0, пока команда Syllable будет работать над следующим значительным шагом, без изменений в ABI и API, так что пользователи смогут устанавливать новые программы и драйверы без обновления системы каждые полгода или около того. Выход 1.0 также будет удачным моментом для расширения рекламы и продвижения.

А что потом? Ну, впереди нелегкий путь, но Syllable могла бы повторить «возрождение *Mozilla*» с её *Firefox*, только среди настольных систем. Новый подход, как в *Mozilla Suite*, был нужен, чтобы сделать Gecko массовым, и, возможно, Syllable может сделать что-то подобное для открытых настольных ОС.

И она может успешно сосуществовать с Linux: все разработчики Syllable согласны с тем, что Linux вполне может остаться на серверах и рабочих станциях разработчиков; но с Syllable, заполняющей нишу домашних систем, мы получили бы сильную и открытую ОС, способную бросить вызов каждому. Нас ожидают несколько любопытных лет.

ЛИЧНО ВАМ

Есть ли место на страницах LXF для альтернатив, подобных Syllable? Или вы считаете это отклонением от основной темы?
Пишите на **letters@linuxformat.ru**

Ресурсы

Наконец, вот список мест, где вы можете получить помощь по Syllable, информацию о разработке и найти новое ПО:

- Официальный сайт: **www.syllable.org**
- Аппаратная поддержка: **www.syllable-hardware.info**
- Учебные пособия: **www.other-space.com/sub**
- Разработка: **http://development.syllable.org**
- Программы: **http://kamidake.other-space.com**

Прежде чем нырнуть в Интернет, исследуйте наш диск: там находятся руководства по разработке, документация, дополнительные программы и многое другое. Да и форум *LXF* – Other OS – всегда открыт для общения на тему Syllable. Дерзайте! **LXF**

ИСТОРИЯ РАЗРАБОТКИ SYLLABLE

- **Середина 1990-х** – Курт Скауен втихомолку приступил к работе над AtheOS
- **Март 2000** – Первый публичный релиз AtheOS
- **Июль 2002** – Ответвление Syllable; первый релиз, 0.4.0
- **Июнь 2003** – 0.4.4, первый релиз с загрузкой с CD
- **Октябрь 2003** – появился 0.5, с поддержкой USB и новой медиа-структурой
- **Сентябрь 2004** – 0.5.4 представил полностью переписанный рабочий стол
- **Декабрь 2005** – 0.6, новая серия с множеством обновлённых драйверов и стабилизацией API

ПРИСОЕДИНЯЙТЕСЬ

Понравилось? Можете поучаствовать в проекте

Мы надеемся, что вы уже познакомились с Syllable и поняли, что предлагает эта система. А сейчас самое время подключиться к активному проекту, который способен радикально изменить будущую ситуацию на рынке настольных систем – здесь есть чем заняться.

- **Продвижение**. Syllable не очень широко известна вне круга любителей открытых систем. Несколько упоминаний на Slashdot и OSNews приподняли её популярность, но многие продолжают считать её одним из дистрибутивов Linux, или не подозревают о её простоте и производительности. Пока Syllable еще не стала массовой системой, люди, рассказывающие о ней и привлекающие новых разработчиков, будут очень полезны.
- **Тестирование**. Команда разработчиков Syllable – одна из самых дружелюбных и прагматичных. Если что-то не работает, отправьте сообщение с описанием проблемы в список рассылки (см. **www.syllable.org**) – и получите помощь и множество идей. На данной стадии разработки хакеры, работающие над Syllable, с благодарностью примут любые сообщения об ошибках и конструктивные предложения по улучшению любимой системы.
- **Разработка**. В чём Syllable больше всего нуждается, так это в разработчиках. Те, кому интересно писать ядро или графический интерфейс, нужны больше всего, но есть огромный спрос и на разработчиков приложений. Что может быть прекраснее участия в молодой ОС, активно продвигающейся в новых направлениях и открывающей новые горизонты! Например, если вы пишете клиент IRC для Linux, ему легко затеряться в море Freshmeat; а напишите его для Syllable, и вы получите одобрение сообщества. Не исключено, что он даже войдёт в состав самой операционки.

Один из лучших способов присоединиться к Syllable – портировать существующие программы. Автор этой статьи портировал эмулятор ПК, *DOSBox*, для баловства, связанного с F1GP.

Человек у руля

Кристиан ван дер Влиет, лидер проекта Syllable, отвечает за общее направление разработки операционной системы. Мы уговорили Вандерса дать небольшое интервью.

LXF: **Что будет в следующих релизах Syllable?**

КвдВ: В выпусках 0.6 мы занимаемся устранением основных брешей в функциональности. Это означает добавление основных функций: поддержка печати, коммутируемых соединений по PPP, OpenGL, гигабитных сетей, Wi-Fi, поддержка таких вещей как dual-head video и прочих функций, которые предполагаются в настольной операционной системе. Компонентам рабочего стола будет уделено особое внимание, по части улучшения удобства работы и добавления новых функций.

Я надеюсь, мы увеличим число приложений для Syllable. Пока ещё требуется заполнить огромные пробелы в функциональности. Мы продолжим улучшение наших приложений для Интернета, а также добавим побольше игр, как только наша поддержка OpenGL и SDL станет более зрелой. Разработчики будут рады узнать, что мы также рассматриваем возможность добавить привязки (binding) для других языков программирования, помимо C++.

LXF: **Что сейчас Syllable требуется больше всего?**

КвдВ: Разработчики, разработчики и ещё раз разработчики! Нам особенно нужны хакеры для работы над ядром и разработчики приложений. Также нужны люди, способные помочь с графическим интерфейсом и другими системными библиотеками, заняться отладкой и выявлением ошибок, портированием и интеграцией программ… Фактически, разработчики нам нужны везде.

Кроме того, нужны люди, способные писать документацию или помогать новым пользователям на форумах и в списках рассылки. Мы будем рады всем, кто хотел бы переводить Syllable и её приложения на другие языки. Мы также не отказываемся от любых пожертвований: мощного полезного оборудования и просто наличных!

LXF: **Настанет ли когда-нибудь год Linux на рабочем столе?**

КвдВ:: Одни думают, что он уже настал, другие твердят это столько, сколько я себя помню; но я, честно говоря, не верю, что это случится. Есть ряд фундаментальных проблем, которые, по моему мнению, делают Linux непривлекательным выбором для разработчиков прикладного ПО и OEM-поставщиков. Без поддержки со стороны производителей такой год никогда не настанет. Доля Linux на рабочих столах пользователей, похоже, будет колебаться в районе отметки 3–4%. Конечно, само по себе изумительно, что открытая операционная система может конкурировать с фирмами вроде Apple, но я не вижу причин, по которым её доля на рынке может вырасти выше указанного предела.

Наши в IBM

Корпорация IBM известна как один из самых крупных игроков на рынке Open Source. Инвестиции «голубого гиганта» в эту отрасль превышают миллиард долларов, а инженеры компании трудятся над множеством открытых проектов – от ядра Linux до J2EE-серверов. IBM неоднократно предоставляла сообществу Open Source и свои собственные наработки, а также осуществляла коммерческую поддержку открытых решений, что очень важно для продвижения Linux в бизнесе. Что будет дальше? Что ожидает Linux на рабочем столе? Как IBM будет развивать открытое ПО в России?

Виктор Федосеев встретился с **Алексеем Широких**, чтобы получить информацию из первых рук.

Визитка LXF

Алексей Широких

Наш бывший соотечественник, ныне руководитель направления Linux/Open Source в быстроразвивающихся странах в корпорации IBM. Проживает в Рачестере, Нью-Йорк.

ИСПОЛЬЗУЕТ UNIX	10 лет
ЯЗЫКОВ ПРОГРАММИРОВАНИЯ	5
КОЛИЧЕСТВО ПК	3,5
ДНЕВНАЯ НОРМА КОФЕ	4-5 чашек
ПАР САНДАЛИЙ	1

ОН ГОВОРИТ «Операционную систему мы делать не будем»

***LINUX FORMAT*: Официально Ваша должность в IBM называется «руководитель направления Linux/Open Source в быстроразвивающихся странах». Расскажите, пожалуйста, что под этим подразумевается? Каковы Ваши должностные обязанности?**

Алексей Широких: Если мы будем идти с самого верха, то первым, кого мы встретим, будет мой начальник – Скотт Хенди (Scott Handy), который занимается стратегией развития Open Source и Linux во всем мире. Следом за ним иду я. Я отвечаю за ту же стратегию, но уже адаптированную к быстроразвивающимся странам. То есть, фактически, я занимаюсь стратегией, маркетингом стратегии, оказываю помощь в создании нужных предложений, которые будут продвигаться в этих регионах. Если говорить про IBM вообще, то таких предложений может быть 100, а тех, что применяются в странах с быстроразвивающимся рынком, скажем, будет всего 5. Я занимаюсь адаптацией предложений для таких стран.

Лично я отвечаю за четыре страны: Индию, Китай, Бразилию, Россию. Но это половина дела. Самое сложное – это исполнение данной стратегии. Одна из моих обязанностей – это понять, что работает в данной стране, как это работает, почему это работает и как мы можем масштабировать эти решения на другие страны, чтобы не затрачивать усилия различных групп для решения одних и тех же проблем.

***LXF:* То есть, по большему счету, Вы занимаетесь выработкой общей стратегии и координацией деятельности местных локальных групп в каждой конкретной стране?**

АШ: Да. Плюс к этому, я занимаюсь и занимался Grid'ами и отвечаю, так сказать, за community of practices в области Grid в Северной Америке. Таким образом, моя задача – продвижение не только Linux, но и других высоких технологий, таких как виртуализация и Grid в быстроразвивающихся странах.

***LXF:* А вы как считаете, ситуация в Linux в других быстроразвивающихся странах сильно отличаются от ситуации с Linux в России?**

АШ: С точки зрения роста?

***LXF:* Нет – с точки зрения особенности рынка**

АШ: С точки зрения особенностей, я бы не сказал, что отличия слишком велики. Что действительно отличается, так это своеобразность, скажем так, индустрии на этих рынках. Через все страны проходит один сектор. Это – правительство. Интерес правительства к Linux, интерес правительства к Open Source. Причем, я бы сказал, не столько к технологии, сколько к открытости. Сложно говорить, что правительство заинтересовано в Linux как в таковом – что оно знает о Linux? У правительства совсем иная сфера деятельности. Их задача – создать платформу, открытую инфраструктуру, которая может масштабироваться. Потому что, в общем случае, мы начинаем с федерального правительства, затем спускаемся на уровень отдельных субъектов федерации, муниципалитетов. Мы должны создать такую инфраструктуру, которая легко масштабируется на различные уровни. И Linux – это, на текущий момент, ключевая и, пожалуй, единственная реальная технология, которая позволяет нам создать такую инфраструктуру.

Когда у нас на руках оказывается нужная инфраструктура, мы едем в каждую страну и начинаем изучать преобладающие в ней интересы. К примеру в России – это газовая и нефтяная отрасли. Это специфика вашей страны. И мы создаем кластеры, используем Grid-технологии и так далее. Если мы, к примеру, переместимся в Бразилию – то это банковская сфера. Что касается Linux, то в данной области очень серьезный вклад в его использование делают мейнфреймы – IBM System z. Причем там происходит такой рост в использовании мейнфреймов под управлением Linux, что это накладывает отпечаток и на другие платформы. Конечно, хорошо, что растет этот сектор и эта платформа. Но с другой стороны, другие платформы от такого роста страдают.

Что касается Индии – это телекоммуникационный бизнес. А с точки зрения головного офиса, мы видим как развиваются региональные сектора, индустрии в этих странах, и мы можем перенимать опыт одной страны и применять его на просторах других стран.

***LXF:* А что на ваш взгляд на текущий момент времени требует пристального внимания в России?**

АШ: Правительство и безопасность. Вы понимаете – это специфика России. Как только мы создаем решение, которое безопасно и которое принято правительством России, то мы можем говорить о нем с банками и с другими органами.

***LXF:* В какой из перечисленных Вами четырех стран более высокая динамика развития Linux? Где Linux-решения IBM имеют больший успех?**

АШ: Общий рост продаж Linux за последний год – более 50%. Это очень существенный рост и динамика. Быстрее всех растет Китай, Индия, Бразилия и потом Россия. Однако если смотреть на потенциал роста, то все эти страны находятся в одной весовой категории. Просто на текущий момент времени объем рынка Linux сильно различается, но динамика при этом схожая. Если в Китае рынок может вырасти с 500 миллионов долларов до 1 миллиарда, то в России он может вырасти с 50 миллионов до 100 миллионов. Однако в процентном исчислении – динамика роста в Китае и России одинаковая.

О МЕСТЕ LINUX В МИРЕ

> «Раньше Linux был технологией. Это было модно, необычно. На текущий момент Linux покупают потому, что это уже круто. »

>>

LXF: А как Вы сами оцениваете рост положения Linux во всем мире?

АШ: Раньше Linux был технологией. Это было модно, необычно. На текущий момент Linux покупают потому, что IBM предоставляет 400 продуктов, которые работают на этой ОС. Это уже круто. На самом деле, основные игроки на рынке предоставляют продукты, которые работают на Linux, то клиенты уже начинают покупать решения, которые служат для выполнения определенных задач. Поэтому Linux уже перешел ту стадию, когда это была чистая технология, позиционировавшаяся как противовес Microsoft и – по совместительству – идеология.

Наступило более спокойное время. Время технических решений, время бизнес-решений, время, когда Linux уже начинает использоваться бизнесом. Однако сейчас еще серьезно чувствуется влияние того времени, когда Linux был чистой технологией и идеологией, которую развивали энтузиасты. Это хорошо, потому что сейчас мы находимся все же в неком промежуточном состоянии – между голой идеологией и чистым бизнесом. И идеология нужна нам для того, чтобы не угас интерес к Linux, чтобы создавались необходимые бизнес-решения, чтобы не было вакуума, а был плавный переход между двумя чистыми состояниями Linux.

На самом деле, в России мы хотели бы сосредоточится на создании большего количества прикладных приложений.

LXF: А чего, на ваш взгляд, сейчас не хватает для бурного развития Linux в России?

АШ: Прикладных приложений, написанных специально для российского клиента.

LXF: Это верно. В настоящий момент идет создание таких приложений. К примеру, фирма «1С» планирует летом представить beta-версию серверной части продукта «1С:Бухгалтерия» для Linux…

АШ: Поживем-увидим. В любом случае, это хорошо для Linux. Но внутри IBM мы глубоко уверены, что нужна сертификация безопасности Linux на соответствие российским стандартам. Это основное. Если у нас будет это, то у нас будет очень интересное и привлекательное решение, с которым мы сможем выйти на правительство. А через него – выйти на другие области российского бизнеса.

LXF: Некоторое время назад, в СМИ прошла информация о том, что компания IBM разорвала контракт на обслуживание своего компьютерного парка с Microsoft и планирует целиком перейти на Linux. Это правда?

АШ: Да, я слышал об этом и мы обсуждали данный вопрос, но это все же не так.

LXF: А вообще, компания IBM в обозримом будущем собирается установить Linux на рабочих столах своих сотрудников?

АШ: Операционную систему мы делать не будем [смеется] У нас есть внутренний Linux-клиент – Client For Business, который основан на Red Hat и находится сейчас в стадии beta-тестирования. Устанавливать его всем и каждому мы пока не планируем. Причина этого проста – все упирается в приложения. Если из 10 человек один работает в Linux и этот человек создает документы, которые потенциально могут не читаться или не так читаться в Windows, то это будет стопорить всю работу. К примеру, этот человек создает документ в OpenOffice.org, другой человек открывает его в Microsoft Word – все замечательно, но съехало форматирование и он тратит на него два часа времени – впустую.

LXF: Существует какой-то план действий по разработке или переводу нуж-

ных внутрикорпоративных приложений IBM под Linux?

АШ: Конечно. Такие стратегии всегда существуют. Над данной задачей работают сотрудники нашей компании, но пока все это находится на стадии пилотных проектов.

LXF: **А у вас есть опытные зоны, где вы обкатываете своего Linux-клиента, так сказать, в деле?**

АШ: Данный клиент может скачать и установить каждый сотрудник компании. На моем ноутбуке, он, к примеру, стоит. В данном клиенте есть почти все, чем пользуются сотрудники внутри IBM: Lotus Notes с подключаемым модулем для Workplace и так далее Но еще раз говорю: все это – бета-версии. Каких-то специализированных зон пока нет. В данный момент, переход на Linux упирается в эффективность работы. Вы понимаете, я могу купить Феррари и настраивать ее 364 дня в году и один день в этом году на ней ездить. А могу купить Жигули и ездить на нем все 365 дней в году.

LXF: **И насколько, по вашему ощущению, Linux готов к применению в качестве настольной ОС?**

АШ: В том, что касается рабочих столов, я бы не хотел вдаваться в подробности. Существует такое определение как «transactional desktop». Скажем, у вас в компании есть секретарь, директор по маркетингу, финансист и так далее. Каждому из них нужны свои приложения. Секретарю не нужны финансовые программы и так далее. Так вот, все движется к тому, что будут распространяться «transactional desktop'ы», оптимизированные под определенную роль.

LXF: **А как вы думаете, через какое время появятся версии дистрибутивов Linux, которые займут ощутимую долю настольного рынка? Я понимаю, что вопрос из области «пальцем в небо», но все же – ваши оценки, ваши ощущения?**

АШ: Вообще делать предсказания в области технологий очень сложно. Взять тот же Google...

LXF: **Назовите, пожалуйста, три причины против использования Linux на рабочем столе прямо сейчас.**

АШ: Малое количество клиентского ПО. Привычка. Наше образование построено на Windows и люди привыкли работать в Windows. Ну и некоторая незаконченность клиентского ПО в плане совместимости с текущими приложениями, которые широко используются. Тот же OpenOffice.org и MS Office. Да – там все уже очень неплохо – но присутствует ощущение некоторой незаконченности OpenOffice в плане совместимости с MS Office.

LXF: **Помимо Linux, на рынке существуют большое количество других открытых ОС, к примеру – FreeBSD. Какова ваша оценка перспективности данных ОС на рынке?**

АШ: Я не знаю. Это классическая бизнес-проблема. Если существует только одна ОС, то это монополия. За это платит клиент. Если существует много ОС – это другая проблема. Потому что очень много всего не может быть качественным. Наступает состояние фрагментированного рынка и клиент становится абсолютно дезориентированным. Поэтому я не думаю, что у данных ОС, в отличие от Linux, есть сколько-нибудь большие перспективы на коммерческом рынке.

LXF: **Скажите, а каким был Ваш первый дистрибутив Linux?**

АШ: Slackware, конечно.

LXF: **Ну и последний вопрос. Что бы вы хотели пожелать нашим читателям?**

АШ: Ваш журнал, как и все подобные журналы в мире, читают много разных людей. Я думаю, и Linux Format, и вашим читателям нужно постепенно уходить от чистых технологий – к примеру, обсуждения программ на Perl, настройки Apache и так далее – нужно двигаться к бизнес-ориентации, не забывая, конечно и технологии. Однако, как я уже говорил, чистая технология – это еще не все. Linux – это уже рынок и притом – быстро растущий. А рынок – это решение конкретных бизнес-проблем. Не нужно смотреть на Linux, как противовес Microsoft. Это уже прошлое. Нужно решать задачи и предлагать решения для бизнеса. LXF

ВОПРОС ЧИТАТЕЛЮ

Что, на ваш взгляд, интереснее: технология или бизнес-решения? Подключайтесь к дискуссии на www.linuxforum.ru

ЗНАКОМСТВО с IBM WEBSPHERE COMMUNITY EDITION

НА ДИСКЕ

В последние годы мы стали свидетелями того, как Open Source проникает в области, где до того правили бал коммерческие и очень дорогие решения. **Александр Цымбал** рассказывает об одном из героев этой революции – J2EE-сервере WebSphere Community Edition, базирующемся на Apache Geronimo.

WebSphere Community Edition (WAS CE) – это компактный и не требующий для работы большого количества ресурсов, бесплатный J2EE-сервер с открытым кодом, разрабатываемый IBM на основе J2EE-сервера Geronimo консорциума Apache. Несмотря на свою «легкость», WAS является полноценным J2EE-сервером, сертифицированным на соответствие спецификации J2EE 1.4. Он создавался не только как прекрасная платформа на разработчиков, которые только начинают использовать J2EE и изучают ее возможности, но и как операционная среда для реальных приложений, обслуживающих малый и средний бизнес. Кроме того, WAS CE является не «изолированным» решением, а полноценным младшим членом семейства WebSphere, что подразумевает простоту перехода на более мощные реализации.

В начале этого года появилась версия 1.0, (после ряда milestone-релизов), которая доступна для скачивания по адресу http://www.ibm.com/developerworks/downloads/ws/wasce/?S_TACT=105AGX28&S_CMP=DLMAIN.

Единственной платной составляющей использования WAS CE для разработки, развертывания, распространения и выполнения приложения является техническая поддержка от IBM – в тех случаях, когда пользователь считает полезным или необходимым получить ее.

WAS CE построен по технологии, впервые использованной для разработки J2EE-сервера приложений с открытым кодом Geronimo (Apache Software Foundation). В основу проекта Geronimo была положена компонентная модель GBeans, реализующая концепцию Inversion of Control (IoC). Часто вместо IoC используется термин «dependency injection», что можно перевести как «включение зависимостей». Хотя для разработчика J2EE-приложений не имеет значения, как именно реализован используемый им J2EE-сервер, знакомство с основами этой концепции и компонентной модели GBeans облегчает понимание смысла некоторых тегов XML-дескрипторов и структуры каталогов, создаваемых на диске в процессе инсталляции.

Модель Gbeans

Среда GBeans является универсальной средой, которая сама по себе никак не связана с J2EE. Как любая компонентная модель, она включает в себя «Контейнер», т.е. среду управления циклом жизни

Модуль J2EE	Стандартный дескриптор	План развертывания Geronimo
EJB JAR	META-INF/ejb-jar.xml	META-INF/openejb-jar.xml
Web-приложение	WEB-INF/web.xml	WEB-INF/geronimo-web.xml
Коннектор J2EE	META-INF/ra.xml	META-INF/geronimo-ra.xml
EAR-приложение	META-INF/application.xml	META-INF/geronimo-application.xml
Клиентское приложение J2EE	META-INF/application-client.xml	META-INF/geronimo-application-client.xml

компонентов (она называется Geronimo Kernel), и собственно компоненты. Компоненты GBeans имеют следующие основные особенности:

- Они могут иметь состояние, которое может храниться в течение срока жизни компонента;
- Компоненты могут сами хранить связи друг с другом;
- Компоненты могут реагировать на определенные собы тия (в этом случае выполняется соответствующая часть бизнес-логики компонента, созданная его разработчиком) .

«Поля состояния» компонента можно разделить на две группы – атрибуты и связи.

Таким образом, система строится только из GBeans-компонентов. Эти компоненты взаимодействуют друг с другом не непосредственно, а посредством ядра Geronimo. Связи между компонентами, называемые «зависимостями» (dependencies), прописываются на уровне XML-описаний компонентов, которые во многих компонентных моделях, например, EJB, обычно называются «дескрипторами развертывания» (deployment descriptors), а в модели GBeans – «планами развертывания» (deployment plans). Ядро Geronimo, анализируя планы, «узнает» о связях компонентов друг с другом и может генерировать динамические proxy-объекты, обеспечивающие связь компонентов, и автоматически добавлять ссылки на создаваемый экземпляр компонента во все компоненты, зависящие от данного.

В процессе установки компоненты, логически связанные друг с другом, объединяются в конфигурации. Для кода компонентов, входящих в конфигурацию, выполняется процесс сериализации, и такая сериализованная конфигурация записывается в специально предназначенный для этого каталог. После этого для запуска конфигурации сериализованное представление конфигурации нужно просто загрузить. Для того, чтобы компонент GBeans можно было найти, с ним сопоставляется уникальное имя. Регистрация имен и их поиск выполняются с использованием стандартных возможностей JMX.

В случае «чистого» компонента GBeans его план развертывания содержит всю необходимую на уровне конфигурации информацию. В проектах Geronimo и WAS CE такие компоненты используются очень широко. Например, таким GBeans-компонентом является реализация сервера JMS или координатора объектных транзакций. Для развертывания GBeans-компонента создается план развертывания, который представляет собой XML-документ.

Если же рассматривать EJB-компоненты или Web-компоненты J2EE, то соответствующие им GBeans-компоненты генерируются на основе кода и дескрипторов, уже существующих для этих сущностей, и план развертывания для GBeans-компонента содержит только дополнительную информацию. Таким образом, при использовании стандартных J2EE-сущностей план развертывания очень удобно трактовать как специфический для реализации J2EE-контейнера дескриптор развертывания, хорошо знакомый всем программистам, имеющим опыт работы с J2EE-серверами (необходимость такого рода дополнительных дескрипторов связана с тем, что спецификации J2EE-дескрипторов просто не оговаривают все необходимые параметры для EJB-компонентов, сервлетов или JSP-документов). Соответственно, планы развертывания будут в чем-то похожи на обычные дескрипторы развертывания для таких J2EE-сущностей, как приложений J2EE (тег <application>), клиентское приложение J2EE (тег <application-client>), web-приложение (тег <web-app>), EJB-компонент (<ejb-jar>) или ресурсный адаптер (тег <connector>). Процедура развертывания таких GBeans-компонентов похожа на процедуру развертывания соответствующих J2EE-элементов. Как правило (хотя это и не обязательно), план развертывания Geronimo включается в J2EE-модуль вместе со стандартными дескрипторами (в каталоги **META-INF** и **WEB-INF**). При таком подходе имена XML-планов развертывания должны быть фиксированными – они приведены в таблице вверху страницы.

Основные элементы WAS CE

Сервер WAS CE полностью использует преимущества компонентной структуры GBeans, и каждый сервис в его составе является отдельным компонентом. Каждый из таких компонентов создан на основе уже существующих и апробированных на практике продуктов с открытым исходным кодом, наиболее важными из которых являются следующие:

- OpenEJB – EJB-контейнер, соответствующий спецификации EJB 2.1;
- Tomcat и Jetty – альтернативные реализации Web-контейнера;
- Active MQ – реализация JMS;
- TranQL – реализация пула соединений JDBC и механизма сохранения состояния EJB entity-компонентов в режиме CMP (Container-Managed Persistence);
- MX4J – реализация JMX
- Axis – реализация Web-сервисов;
- Derby – реляционная СУБД.

При старте сервера пользователь может запускать только те сервисы, которые необходимы в каждом конкретном случае.

Для запуска сервера на платформах Linux и Windows требуется минимум 120 Мб дискового пространства (не считая JDK) и не менее 128 Мб RAM (желательно 256 Мб). Поддерживаются JDK 1.4.2 от Sun и IBM.

Запуск и настройка WAS CE

WAS CE и Geronimo могут поставляться (в бинарном формате, а не в виде исходного кода) как в форме дистрибутива, так и обычного архива.

Наиболее удобен первый способ – с использованием программы установки, которая (начиная со следующей версии) позволит в интерактивном режиме сконфигурировать сервер и настроить многие параметры его функционирования. Для текущей версии программа установки просто запрашивает, согласны ли вы с лицензионным и соглашением и просит указать каталог для инсталляции. Помимо этого, программа установки проверяет наличие JDK на вашем компьютере и заносит соответствующую информацию в командные файлы для запуска сервера.

Во втором случае архив нужно просто распаковать в выбранный каталог (каталог установки).

Выбор одного из двух поддерживаемых вариантов JDK – от IBM или от Sun – остается на усмотрение пользователя.

После установки в указанном каталоге создаются следующие основные подкаталоги:

- **/bin**, который содержит исполняемый код сервера (файл **server.jar**), средства для развертывания приложений (**deploy.jar**), утилиты для запуска клиентских приложений в формате J2EE (**client.jar**) и остановки сервера (**shutdown.jar**), а также командные файлы для запуска сервера на платформах Linux (**startup.sh**) и Windows (**startup.bat**).
- **/config-store** – содержит подкаталоги модулей WAS CE – модулей в смысле технологии GBeans и Geronimo, а не J2EE. Модулями в таком понимании являются приложения J2EE, коннекторы Java, отдельные объекты, например, пул соединений с БД, сервисы – например, реализация брокера сообщений JMS. Некоторые из находящихся в данном каталоге модулей необходимы для работы J2EE-сервера и ядра Geronimo и запускаются всегда (таких в этом каталоге немного), некоторые запускаются тогда, когда это необходимо пользователю. Именно в этот каталог помещаются конфигурации, соответствующие приложениям, созданным прикладным программистом. Каждый подкаталог данного каталога имеет порядковый номер, а список модулей, содержащий эти номера (и имена JMX) находится в файле **index.properties** в этом же каталоге. Этот список обновляется автоматически.

>>

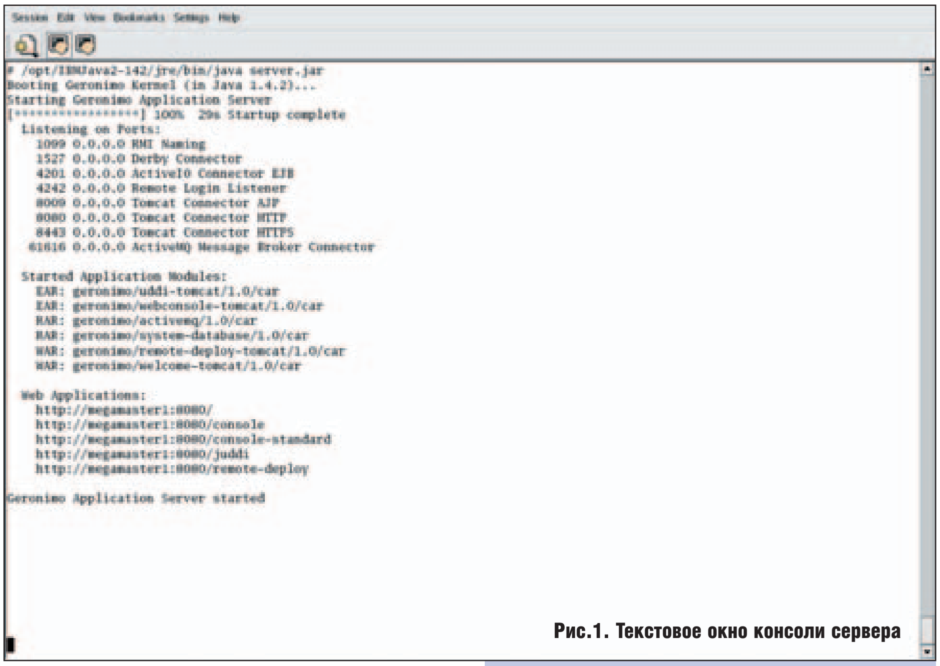


Рис.1. Текстовое окно консоли сервера

- **/lib** – содержит (в виде jar-файлов) реализацию ядра Geronimo и основных сервисов.
- **/repository** – этот каталог можно трактовать как набор библиотек – как системных, так и прикладных. Разработчик помещает в этот каталог (точнее, в один из ее подкаталогов) объекты и компоненты, которые должны быть доступны для различных модулей и приложений, развернутых на сервере. Например, сюда нужно помещать jar-файл, который содержит классы JDBC-драйверов для конкретных БД. Как правило, данные модули загружаются не непосредственно сервером, а косвенно, посредством загрузки других модулей, которые, в свою очередь, используют модули каталога **repository**.
- **/schema** – этот каталог содержит XML-схемы и DTD для используемых при работе сервера классов XML-документов.
- **/license** – содержит текст лицензии для WAS CE на различных языках. Текст лицензии на русском должен появиться в апреле 2006 г.
- **/var** – каталог, содержащий информацию о настройках сервера и его отдельных сервисов. Здесь находится основной файл настройки параметров сервера – **/var/config/config.xml**.

Поскольку сервер представляет собой, главным образом, набор сериализованных конфигураций, состоящих из GBeans-компонентов (которые имеют атрибуты), то поведение сервера можно менять при каждом запуске, изменяя набор конфигураций или задавая различные значения атрибутов компонентов. Список конфигураций и значения атрибутов их компонентов хранятся в файле **<каталог_установки>/var/config/config.xml**. Пользователь может менять содержимое этого файла – либо явно, с помощью любого текстового редактора (сервер при этом не должен быть запущен), либо неявно, устанавливая те или иные компоненты с помощью утилиты deploy или работая с консолью администратора.

Изменение атрибутов компонентов в файле config.xml – самый удобный способ менять настройки сервера, например, номера портов, которые используются при работе тех или иных сервисов WAS CE.

Запуск сервера

Запустить сервер в той конфигурации, в которой он был запущен прошлый раз (при первом запуске – с конфигурацией по умолчанию) можно, выполнив командный файл **startup.bat** (**startup.sh** для Linux), не указывая никаких параметров. По мере запуска в текстовое окно консоли сервера (не путать с графической консолью администратора) выводятся сообщения о запущенных конфигурациях и некоторых их параметрах – например, номерах используемых портов.

Для изменения конфигурации сервера удобнее всего использовать утилиту развертывания компонентов (она кратко описана в следующем разделе), а также консоль администратора сервера.

После запуска сервера с конфигурацией по умолчанию создается пул соединений с реляционной БД Derby (бывшая IBM Cloudscape), поэтому пользователь может сразу приступить к работе с этой БД. Консоль администратора сервера содержит страницу, на которой пользователь может задавать SQL-операторы для работы с выбранной БД и просматривать результаты выполнения этих запросов.

Остановить сервер можно различными способами – из консоли администратора, с помощью командного файла **bin/shutdown.bat** (**shutdown.sh** для Linux) или просто нажатием CTRL-C в окне консоли сервера. Перед остановкой сервера его текущая конфигурация записывается в файл **<каталог_установки>/var/config/config.xml**.

Утилита развертывания компонентов

Для управления конфигурациями сервера используется утилита deploy (загрузчик).

Загрузчик используется для выполнения различных действий, основными из которых являются:

- Загрузка конфигурации (команда deploy);
- Выгрузка конфигурации (undeploy);
- Остановка конфигурации (stop);
- Запуск конфигурации (start);
- Получение списка конфигураций (list-modules)

В терминах J2EE под «конфигурацией» часто понимается то, что обычно называют «модулем».

Список и описание команд можно получить с помощью следующей командной строки:

```
java –jar deployer help [команда]
```

или просто выполнив командный файл **/bin/deploy.bat(sh)** без указания параметров

Для выполнения большинства команд загрузчика необходимо указывать имя пользователя и пароль. По умолчанию используются следующие значения:

```
user=system
password=manager
```

Если эти параметры не указаны в командной строке, то они будут запрошены в интерактивном

Рис.2. Консоль администратора

режиме. Для указания этих параметров в командной строке используйте следующие опции:

```
java –jar deployer –user имя_пользователя
–password пароль …
```

Состав групп пользователей и параметры пользователей находятся в файлах groups.properties и users.properties, соответственно, которые размещены в каталоге **<install_dir>/var/security**.

Команды могут выполняться по-разному – в зависимости от того, запущен сервер или нет. В большинстве случаев требуется, чтобы сервер был запущен.

В общем случае требуется указать сервер, в котором развертывается модуль (или модули), модуль (по его глобальному имени) и план развертывания модуля, если он необходим.

Во многих случаях для выгрузки, запуска и установки модулей удобнее использовать консоль администратора.

Консоль WAS CE

Для управления сервером в процессе работы и управления конфигурациями используется консоль администратора. Консоль является серверным приложением, реализована с использованием портлетов, и ее визуальный интерфейс доступен с использованием стандартных web-браузеров, сервлетов и протокола http. Для начала работы с консолью нужно направить браузер по адресу:

http://<хост>:8080/console

Для доступа к функциональности консоли нужно ввести идентификатор и пароль. Сервер поставляется с готовым вариантом реализации системы безопасности, в том числе процедуры аутентификации, и с «администратором» по умолчанию, идентификатор которого – «system», а пароль – «manager».

Начальная страница консоли администратора выглядит как на рис.2:

Создание J2EE-приложений для WAS CE

Создание J2EE-приложений требует от программиста основательных знаний технологий, входящих в состав J2EE, таких, как JNDI, JDBC, JSP, EJB, JCA, JACC, XML-технологий и других. Специфика использования WAS CE – как и любой другой реализации спецификации J2EE – связана с созданием специфических для каждой такой реализации дескрипторов развертывания, которые дополняют обязательные и стандартные дескрипторы развертывания для различных видов компонентов и приложений. В простейших случаях дополнительные дескрипторы развертывания либо вообще не содержат, либо содержат очень немного специфической для данного сервера информации. Кроме того, обычно для наиболее широко используемых инструментов разработки – таких, например, как Eclipse – созданы инструменты, которые автоматически создают (полностью или частично) как стандартные, так и дополнительные дескрипторы развертывания. В частности, WAS CE содержит расширения для Eclipse 3.1.2. Практика показывает, что обычно программисты не испытывают сколько-нибудь серьезных трудностей с указанием параметров в специфических для сервера дескрипторах развертывания.

Основными особенностями WAS CE, на которые стоит обратить внимание при создании J2EE-приложений, являются следующие:

- Для модулей J2EE создаются компоненты GBeans и их конфигурации. Имя конфигурации является уникальным идентификатором этой конфигурации на сервере. Программист в случае необходимости может использовать JMX для поиска соответствующей конфигурации по ее имени, но обычно имена конфигурации указываются при работе с утилитой *deploy* – например, при выгрузке или остановке конфигурации. Имя конфигурации задается в плане развертывания (специфическом для WAS CE дескрипторе развертывания) с помощью атрибута configId.
- Конфигурации хранятся в сериализованном виде в своих подкаталогах каталога **config-store**.
- Как правило, для каждой конфигурации используется собственный загрузчик классов Java.
- Каждая конфигурация может явно задать зависимость от другой конфигурации – для этого используется атрибут parentId. Ядро Geronimo обеспечивает загрузку сериализованного представления «родительской» конфигурации перед загрузкой той конфигурации, которая от нее зависит. С помощью атрибута parentId можно задать явную зависимость только от одной конфигурации. Другой способ задания явной зависимости от других конфигураций – использование в плане развертывания WAS CE тега <import>. Существуют и другие (так называемые «неявные») способы задания зависимостей конфигураций друг от друга. Если разработчик может включить все необходимые ресурсы в одно приложение, то следует поступать именно так, создавая приложения в виде EAR-архивов. В этом случае нет необходимости заботиться о явной или неявной зависимости конфигураций на уровне планов развертывания WAS CE.
- Все параметры времени развертывания приложения, которые необходимы для работы приложения, но не оговорены на уровне стандартных XML-дескрипторов развертывания (информация в JNDI, параметры JDBC-драйверов, объектов JMS, детали реализации абстрактных схем entity-компонентов в режиме CMP и др.), помещаются в планы развертывания WAS CE.
- Информация в планах развертывания WAS CE иногда может переопределять информацию (особенно это касается связей между компонентами), заданную на уровне стандартных дескрипторов.

Простейшее web-приложение

В качестве простейшего примера лучше всего использовать JSP-документ – это позволяет для демонстрации использования сервера задействовать (помимо средств, входящих в состав WAS CE) только текстовый редактор. В качестве клиента выступает стандартный браузер.

Это приложение, в соответствие со сложившейся традицией, в ответ на общение к нему будет отображать на экране броузера строку «Hello World!».

Для создания такого приложения нужно:

- Создать в текстовом редакторе JSP-документ примерно такого вида:

```
<html>
<body>
<h2>Hello World!</h2>
</body>
</html>
```

и сохранить его в файле с именем **index.jsp**

- Создать (например, с помощью текстового редактора) стандартный xml-дескриптор для web-приложения J2EE. Этот дескриптор, в соответствие с требованиями спецификации J2EE, должен иметь имя **web.xml**. Этот дескриптор может выглядеть, например, так:

```
<!DOCTYPE web-app PUBLIC
 «-//Sun Microsystems, Inc.//DTD Web Application
2.3//EN»
 «http://java.sun.com/dtd/web-app_2_3.dtd» >
<web-app>
 <display-name>Archetype Created Web Application</
display-name>
</web-app>
```

В нем нет ничего, кроме стандартного заголовка.

Как и всякий J2EE-сервер, в общем случае, для установки компонента WAS CE требует наличия дополнительного, специфического для этого сервера дескриптора. В данном простейшем случае никакой необходимости в этом нет.

- Необходимо создать архив с расширением WAR, в котором будут находиться:
  – JSP-документ – в «корневом каталоге»
  – Дескриптор **web.xml** – в подкаталоге с именем **WEB-INF**

Для создания такого архива удобно использовать стандартную утилиту jar. Пусть файл архива будет называться **my_first_webapp.war**.

Приложение готово. Осталось запустить сервер и выполнить команду установки WAR-архива – т.е. web-приложения – на этот сервер. Команда может выглядит следующим образом (в одну строку):

```
java -jar D:\AppServerCommunityEdition\bin\deployer.
jar --user system --password manager deploy
my_first_webapp.war
```

Теперь можно обратиться к нашему web-приложению, задав в браузере следующий URL:

http://localhost:8080/my_first_webapp

В данном случае созданное web-приложение получило имя по имени архива. URL не содержит имени самого JSP-документа, так как его имя – **index.jsp** – используется по умолчанию.

Заключение

WAS CE является простой в использовании, хотя достаточно мощной и полностью соответствующей спецификации средой развертывания и выполнения J2EE-приложений, которая прекрасно подходит как для изучения комплекса технологий J2EE, так и создания реальных информационных систем в интересах малого и среднего бизнеса. LXF

ДРАЙВЕР СЕТЕВОГО УСТРОЙСТВА – своими руками

ЧАСТЬ 1 Не нашли, как включить нужную функцию в make menuconfig? Не беда – Игорь Тимошенко расскажет, как написать драйвер самостоятельно!

Эта статья предназначена для тех, кто желает приобрести начальные знания, позволяющие самостоятельно создавать драйверы сетевых устройств, работающих в среде ОС Linux. Не так давно у меня самого возникла такая задача, и я обнаружил явный недостаток информации по этой теме. В доступной литературе на русском языке подробно описаны все драйвера, кроме сетевых. В англоязычной библии на все времена «Linux Device Drivers» (**http://www.oreilly.com/catalog/linuxdrive3**) есть всё, но понимания этого «всего» можно достичь, пройдя не самый простой путь. Отчаянные метания по форумам привели меня к неутешительному выводу, что интересующихся этим вопросом несколько больше, чем что-то понимающих в нём. Итак, в результате многочисленных проб и ошибок, я получил некоторое количество полезных знаний и драйвер, который сейчас успешно использую дома для подключения второго компьютера к Интернету. Простота получившейся программы натолкнула меня на мысль, что путь, который я прошёл, мог бы быть гораздо короче и куда менее извилист, если бы в начале мне удалось прочитать статью, которую я сейчас вам и предлагаю.

Не смотря на то, что по ходу изложения, я постараюсь сделать необходимые разъяснения и ссылки, для успешного и правильного понимания материала статьи, читателю понадобятся некоторые начальные познания. Прежде всего – знакомство с языком программирования С, кроме того, я полагаю, что читатель представляет, как работают компьютерные сети, уже работал в ОС Linux и знает, что такое «модуль ядра».

Все примеры, приведённые в статье – рабочие, и представляют собой необходимые начальные этапы того самого пути, который нужно пройти, чтобы научиться создавать драйверы различных сетевых устройств. Я использовал дистрибутив Mandrake 10.1 с ядром 2.6.8.1-12mdk. Для других версий ядра, возможно, понадобится внести некоторые незначительные изменения в исходные тексты.

Чтобы иметь возможность компилировать модули ядра, после обычной установки дистрибутива, нужно дополнительно установить исходные тексты вашей версии ядра в каталог **usr/src**. В моём дистрибутиве это пришлось сделать вручную, потому, что даже при выборе расширенного варианта установки и указании на необходимость включения в неё исходных текстов, в нужном каталоге у меня оказались только дремучие исходники ядра 2.4. Нужные исходные тексты находятся на третьем диске дистрибутива в папке **/media/main3/** в виде rpm-пакета: **kernel-source-2.6-2.6.8.1-12mdk.i586.rpm**. Их можно установить с помощью команды:

```
rpm -ivh ./ kernel-source-2.6-2.6.8.1-12mdk.i586.rpm.
```

Установку следует производить в консольном режиме из каталога, содержащего пакет, в режиме суперпользователя.

Перед началом работы советую создать свой рабочий каталог **/home/user/<что-нибудь>**, в котором будут находиться ваши исходные тексты и поместить туда Makefile, взятый мной из примера, предлагаемого средой разработки KDevelop, слегка модифицированный, для обеспечения его работоспособности (ох уж эти особенности свободно распространяемых программ!):

```
TARGET = myname
OBJS = myname.o
MDIR = drivers/misc
EXTRA_CFLAGS = -DEXPORT_SYMTAB
CURRENT = $(shell uname -r)
KDIR = /lib/modules/$(CURRENT)/build
PWD = $(shell pwd)
DEST = /lib/modules/$(CURRENT)/kernel/$(MDIR)
obj-m   := $(TARGET).o
default:
 make -C $(KDIR) SUBDIRS=$(PWD) modules
$(TARGET).o: $(OBJS)
 $(LD) $(LD_RFLAG) -r -o $@ $(OBJS)
ifneq (,$(findstring 2.4.,$(CURRENT)))
install:
 su -c «cp -v $(TARGET).o $(DEST) && /sbin/depmod -a»
else
install:
 su -c «cp -v $(TARGET).ko $(DEST) && /sbin/depmod -a»
endif
clean:
 -rm -f *.o *.ko .*.cmd .*.flags *.mod.c
-include $(KDIR)/Rules.make
```

Чтобы приспособить этот файл под свои нужды, замените **myname** на имя вашего модуля. Сборка осуществляется командой **make**.

Добро пожаловать в мир хакеров Linux!

Для начала, создадим простейший модуль ядра, который будет регистрироваться в системе и сообщать об этом миру доступными ему средствами: «Hello world!». Это необходимый шаг: надо же уважать традиции! Кроме этого, он позволяет удостовериться в работоспособности инструментов (компилятора, компоновщика, ...), корректности make-файлов и правильности установки исходных текстов Linux. Кстати, корифеи от ядра настоятельно рекомендуют разрабатывать и отлаживать модули в текстовой консоли. Вызов **printk** (речь о котором пойдет ниже) не работает с графическими терминалами типа **xterm**, поэтому единственным способом понять, что происходит, остается просмотр файлов журнала, а это не всегда удобно. В общем, вооружайтесь люби-

мым текстовым редактором (от *mcedit* до *joe*) и – в путь!

```
/************************************************************
 * Igor Timoshenko (timigor@yandex.ru)                   *
 * Simple module                          *
 ************************************************************/
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
/*============ИНИЦИАЛИЗАЦИЯ МОДУЛЯ=========*/
int ssl_init_module (void)
{
printk(KERN_WARNING «Hello World!.\n»);
return 0;
}
/*============ВЫГРУЗКА МОДУЛЯ===============*/
void ssl_cleanup (void)
{
printk (KERN_WARNING «Goodbye world.\n»);
return;
}
module_init (ssl_init_module);
module_exit (ssl_cleanup);
```

Сохраните этот модуль в каком-нибудь файле (например, **helloworld.c**), внесите соответствующие изменения в Makefile (*см. выше*), а затем дайте команду make. На экране должен появиться текст:

```
make -C /lib/modules/2.6.8.1-12mdk/build SUBDIRS=/home/STATYA modules
make[1]: Entering directory `/usr/src/linux-2.6.8.1-12mdk'
 CC [M] /home/STATYA/helloworld.o
 Building modules, stage 2.
 MODPOST
 CC   /home/STATYA/helloworld.mod.o
 LD [M] /home/STATYA/helloworld.ko
make[1]: Leaving directory `/usr/src/linux-2.6.8.1-12mdk'
```

После успешной компиляции в вашем каталоге образуется множество файлов, из которых для нас интересен собственно объектный модуль **helloworld.ko**. Для загрузки получившегося модуля в ядро нужно использовать команду:

```
insmod ./helloworld.ko
```

При этом наш модуль будет занесён в список загруженных модулей в файле **/proc/modules** (в этом можно убедиться, используя команду **cat /proc/modules | grep helloworld)**. Кроме того, будет запущена инициализирующая функция **ssl_init_module(void)**, вызывающая системную функцию **printk()**, которая предназначена для регистрации событий и предупреждений из режима ядра. Мы не будем сейчас подробно рассматривать работу этой функции, отметим лишь, что переданное ей в качестве аргумента сообщение «Hello World!» после загрузки модуля следует искать в файле **/var/log/messages**, куда (при соответствующей настройке демона **syslog)** отправляется большинство сообщений, приходящих из ядра. Для удобства наблюдения за появляющимися новыми сообщениями в этом файле можно воспользоваться командой

```
tail -f /var/log/messages
```

Если вы проигнорировали советы бывалых и все-таки разрабатываете модули ядра в графической среде, её удобно запускать в отдельном окне терминала, создав своеобразную «консоль сообщений».

Выгрузить модуль из ядра можно при помощи команды:

```
rmmod helloworld
```

При этом в запущенной консоли сообщений можно будет наблюдать, как модуль попрощается с миром все теми же «доступными ему средствами».

Матрица: подключение

Следующим шагом на нашем пути будет создание простейшего модуля сетевого драйвера. Для этого в нашем модуле нужно определить структуру данных сетевого драйвера **struct net_device**, определение которой находится в заголовочном файле **/usr/include/linux/netdevice.h**. В структуре представлено множество функций и полей, из которых для нас наиболее важными являются следующие:

- **char name[IFNAMSIZ]** – содержит имя интерфейса, которое будет отображаться при конфигурировании сетевой подсистемы.
- **int (*open) (struct net_device *dev)** – функция, которая запускается всякий раз, когда утилита ifconfig активирует интерфейс.
- **int (*stop) (struct net_device *dev)** – функция, которая запускается при остановке интерфейса.
- **int (*hard_start_xmit) (struct sk_buff *skb, struct net_device *dev)** – функция, которая активизируется сетевой подсистемой всякий раз, когда нужно передать пакет данных.
- **struct net_device_stats* (*get_stats)(struct net_device *dev)** – функция, которая вызывается всякий раз, когда какое либо приложение пытается получить статистические данные о работе интерфейса.
- **void *priv** – дополнительный указатель, который можно использовать произвольным образом.

Остальные поля и функции обеспечивают драйверу дополнительные возможности, которые мы сейчас рассматривать не будем. Их

>>

ДРАЙВЕРА LINUX: КУРС МОЛОДОГО БОЙЦА

Драйвер устройства – это программа, работающая непосредственно с устройством и позволяющая другим программам общаться с ним через какой-то стандартный интерфейс.

В Linux (в отличие, например, от MS Windows) все драйвера располагаются в пространстве ядра и представляют собой стандартные модули. Такие модули могут быть встроенными в ядро при компиляции или динамически загружаться во время работы ОС. Загружаемые модули более удобны в использовании, т.к. они могут подгружаются только при подключении устройства и не потребляют системные ресурсы при его отсутствии. Для загрузки модуля в ядро и его выгрузки пользователем предусмотрены специальные команды, для программного управления модулями – специальные функции. Помимо «универсального модуля», в ОС Linux существует несколько специализированных типов модулей ядра – это драйвера символьных устройств (принтер, звуковая карта), блочных устройств (дисковые накопители) и сетевых устройств (сетевые карты).

Каждый модуль представлен в ядре специальной структурой данных, состоящей из указателей на его параметры и функции. Набор функций, их расположение в структуре и набор передаваемых ими параметров определены типом модуля и не могут быть произвольно изменены. Программист может лишь наполнить эти функции своим содержимым, чтобы модуль мог выполнять необходимые действия. После загрузки модуль регистрируется в системе при помощи функции, указанной макроопределением **module_init ()**. Далее, ядро вызывает другие его функции, когда какая-либо программа пытается выполнить стандартные действия с устройством, которое обслуживает данный модуль. При выгрузке вызывается функция, указанная макроопределением **module_exit()** и аннулирующая регистрацию модуля в системе.

Модуль драйвера сетевого устройства – это программа, которая работает в составе сетевой подсистемы ОС Linux и предоставляет стандартный программный интерфейс для доступа к аппаратным устройствам протоколам верхнего уровня. Обычно такими устройствами бывают сетевые Ethrnet-карты или модемы. Хорошо реализованный драйвер может работать сразу с несколькими однотипными устройствами. При загрузке в память такой сетевой драйвер запрашивает необходимые для работы ресурсы – имеющиеся сетевые устройства и соответствующие им порты ввода/вывода и прерывания. Для каждого обнаруженного сетевого устройства драйвер размещает в памяти свою структуру данных и заносит её в глобальный список сетевых устройств. Более простые драйверы могут работать только с одним устройством, представленным в ОС его зарегистрированной структурой.

назначение можно узнать в справочной литературе. Следует сразу отметить, что приём данных от сетевого устройства обычно происходит по прерыванию, его обработчиком, и явного указания на этот обработчик в структуре интерфейса не содержится.

Нам нужно заполнить необходимые для работы драйвера поля. Если поле содержит указатель на функцию – необходимо реализовать эту функцию. В нашем драйвере мы определим функции инициализации (**open**), остановки (**stop**) и передачи данных (**hard_start_xmit**), кроме того определим структуру **net_device_stats**, содержимое которой мы будем получать через дополнительный указатель (priv).

Для того, чтобы система узнала, что загружаемый модуль является драйвером сетевого устройства, при загрузке модуль должен сообщить ей об этом специальной функцией **register_netdev()**. О своей выгрузке модуль сообщает функцией **unregister_netdev()**. В качестве аргумента для этих функций передаётся указатель на структуру данных драйвера.

Для инициализации интерфейса в сетевой подсистеме драйвер должен воспользоваться специальной функцией **netif_start_queue ()**, при остановке – функцией **netif_stop_queue()**, аргументом которых также является указатель на структуру драйвера.

При необходимости передачи данных через сетевой интерфейс ОС вызывает функцию **hard_start_xmit()** которой в качестве аргумента передаётся указатель на буфер **(sk_buff)**, содержащий готовую IP-датаграмму. Вызванная функция должна сделать всё необходимое для отправки датаграммы сетевому устройству для передачи на физическом уровне, а затем сообщить системе о том, что пакет передан, вызвав функцию **dev_kfree_skb()**, аргументом которой служит указатель на буфер, полученный от ОС. На этом цикл передачи для драйвера завершается, система может заполнять освобождённый буфер новыми данными, а драйвер ожидает следующей передачи и указателя на готовый буфер.

Поскольку наш драйвер пока не связан с каким либо устройством, функция передачи будет переносить передаваемые данные в свой собственный буфер **(my_buf)**, а затем регистрировать эти данные в виде сообщений при помощи функции **printk()**. Для представления регистрируемых данных в текстовом формате мы введём в программу вспомогательные функции **tpdumpk()** и **printAddr()**.

Приведём текст нашего, пусть простейшего, но уже сетевого драйвера:

```
/*************************************************************
 *  Igor Timoshenko (timigor@yandex.ru)                    *
 *  Simple net driver                               *
 ************************************************************/
#include <linux/module.h>
#include <linux/kernel.h>
#include <linux/init.h>
#include <linux/netdevice.h>
#include <linux/skbuff.h>
/*Структура заголовка IP пакета*/
struct iphdr {
  unsigned int ihl:4;
  unsigned int version:4;
  u_int8_t tos;
  u_int16_t tot_len;
  u_int16_t id;
  u_int16_t frag_off;
  u_int8_t ttl;
  u_int8_t protocol;
  u_int16_t check;
  unsigned char saddr[4]; /* IP address of originator */
  unsigned char daddr[4]; /* IP address of destination */
  /*The options start here. */
 };
/*Структура буфера данных */
struct my_buf {
int tlen;
          union tbuffer{
                    unsigned char tbuff[1024];
                    struct iphdr thdr;
          } tb;
};
struct net_device_stats *stats;
struct my_buf xbf;
int retval, shift, i;

int ssl_stop (struct net_device *dev);
int ssl_open (struct net_device *dev);
static int ssl_xmit(struct sk_buff *skb, struct net_device *dev);
void tpdumpk(struct my_buf *txbf);

/*Структура данных драйвера*/
struct net_device ssl_dev = {
          .name                    = «ssl»,
          .open                    = ssl_open,
          .stop                    = ssl_stop,
          .hard_start_xmit         = ssl_xmit,
};
static struct net_device_stats *ssl_get_stats(struct net_device *dev)
{ return dev->priv; }
/*============ПЕРЕДАЧА ПАКЕТА============*/
static int ssl_xmit (struct sk_buff *skb, struct net_device *dev)
{
          struct my_buf *qbf;
          qbf = &xbf;

          if (skb->len > 1024 – 3) {
                    printk(KERN_WARNING «ssl: T_buffer is small,
dropping packet.\n»);
                    stats->tx_dropped++;
          } else {
          xbf.tlen = skb->len;
          shift = 0;
                    while (shift + 1 <= skb->len) {
                    xbf.tb.tbuff[shift] = skb->data[shift];
                    shift++;
                    }
          qbf->tlen = skb->len;
          stats->tx_bytes += skb->len;
          stats->tx_packets++;
          }
          tpdumpk(&xbf);  //Дамп переданного пакета из буфера
          dev_kfree_skb(skb);
          stats->tx_bytes += skb->len;
          stats->tx_packets++;
          return 0;
}
/*============ИНИЦИАЛИЗАЦИЯ ИНТЕРФЕЙСА ============*/
int ssl_open (struct net_device *dev)
{
          printk(KERN_WARNING «ssl: ssl_open called.\n»);
          netif_start_queue (dev);
          return 0;
}
/*============ОСТАНОВКА ИНТЕРФЕЙСА ============*/
int ssl_stop (struct net_device *dev)
{
          printk (KERN_WARNING «ssl: ssl_stop called.\n»);
          netif_stop_queue(dev);
          return 0;
}
/*=============ИНИЦИАЛИЗАЦИЯ МОДУЛЯ============*/
int ssl_init_module (void)
{
```

```
		stats = kmalloc(sizeof(struct net_device_stats), GFP_KERNEL);
		if (stats) {
		memset(stats, 0, sizeof(struct net_device_stats));
				ssl_dev.priv = stats;
				ssl_dev.get_stats = &ssl_get_stats;
		}
		if ((retval = register_netdev (&ssl_dev))) {
		printk (KERN_WARNING «ssl: Error %d while initializing module.\n»,retval);
				return retval;
		}
		printk(KERN_WARNING «ssl: initializing the module.\n»);
		return 0;
}
/*============ВЫГРУЗКА МОДУЛЯ==============*/
void ssl_cleanup (void)
{
		unregister_netdev (&ssl_dev);
		printk (KERN_WARNING «ssl: Cleaning Up the Module.\n»);
		return;
}
module_init (ssl_init_module);
module_exit (ssl_cleanup);

/*============ ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ ============*/
// Печать IP-адреса в формате «разделенный точками».
void printAddr(unsigned char *addr)
{
int j;
		for ( j = 0; j < 4; j++ )
		{
		printk(«%d», addr[j]);
		if ( j < 3 )
				printk(«.»);
		}
}
// Дамп буфера передатчика
void tpdumpk(struct my_buf *txbf)
{
int i;
		printk(«\n---TRANSMITING---»);
		for ( i = 0; i < txbf->tlen; i++ )
		{
				if ( !(i & 15) ) printk(«\n%04X: «, i);
				printk(«%02X «, ((unsigned char*)txbf->tb.tbuff)[i]);
		}
		printk(«\n»);
		printk(«IPv%d: hdr-size=%d pkt-size=%d protocol=%d TTL=%d «,
				txbf->tb.thdr.version, txbf->tb.thdr.ihl*4, ntohs(txbf->tb.thdr.tot_len),
				txbf->tb.thdr.protocol, txbf->tb.thdr.ttl);
		printk(«\n»);
		printk(«rst= «);
		printAddr(txbf->tb.thdr.saddr);
		printk(« dst= «);
		printAddr(txbf->tb.thdr.daddr);
		printk(«\n»);
}
```

Для загрузки и инициализации скомпилированного драйвера удобно пользоваться небольшим сценарием, текст которого приведен ниже:

```
#!/bin/sh
insmod ./smpldriver.ko
sleep 1
ifconfig ssl 192.168.0.1 up
ifconfig ssl netmask 255.255.255.0
ifconfig
```

Задержка (sleep 1) требуется для того, чтобы система успела зарегистрировать модуль и была готова к инициализации его интерфейса к моменту подачи команды **ifconfig**. После загрузки и инициализации нашего драйвера на экране должен появиться примерно такой текст (информация о других сетевых интерфейсах опущена для экономии места):

```
ssl	Link encap:AMPR NET/ROM HWaddr
	inet addr:192.168.0.1 Mask:255.255.255.0
	UP RUNNING MTU:0 Metric:1
	RX packets:0 errors:0 dropped:0 overruns:0 frame:0
	TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
	collisions:0 txqueuelen:0
	RX bytes:0 (0.0 b) TX bytes:0 (0.0 b)
```

Как видно, наш интерфейс запущен, имеет собственный IP-адрес и готов к работе.

Если ваш компьютер подключен к Интернету, то перед тем, как отправлять на наш интерфейс данные, убедитесь, что этому не препятствуют настройки компьютера. Сконфигурируйте межсетевой экран (**firewall**) с учётом появления новой подсети, или даже вообще отключите его на время отладки (особо осторожные могут одновременно отключиться и от Интернета).

Для проверки работоспособности нашего интерфейса нужно запустить в отдельной консоли команду **tail -f /var/log/messages** для отображения текущих сообщений из ядра, а в вашей рабочей консоли воспользоваться командой:

```
ping 192.168.0.2.
```

Таким образом, мы пытаемся отправлять ICMP-пакеты на несуществующий интерфейс, который доступен через наш драйвер. Поскольку такого интерфейса, как и функций приема данных, у нашего драйвера пока нет, для ОС все отправленные пакеты будут потеряны. Но не для нас! Все передаваемые пакеты будут документироваться в файле **/var/log/messages**, и это можно наблюдать в консоли сообщений примерно в таком виде:

```
Mar 19 09:28:21 eng-fxih kernel: ---TRANSMITTING---
Mar 19 09:28:21 eng-fxih kernel: 0000: 45 00 00 54 00 02 00 00 40 01 F9 53 C0 A8 00 01
Mar 19 09:28:21 eng-fxih kernel: 0010: C0 A8 00 02 08 00 5A 64 3A 45 00 03 05 FA 1C 44
Mar 19 09:28:21 eng-fxih kernel: 0020: 4E 12 08 00 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13
Mar 19 09:28:21 eng-fxih kernel: 0030: 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23
Mar 19 09:28:21 eng-fxih kernel: 0040: 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33
Mar 19 09:28:21 eng-fxih kernel: 0050: 34 35 36 37
Mar 19 09:28:21 eng-fxih kernel: IPv4: hdr-size=20 pkt-size=84 protocol=1 TTL=64
Mar 19 09:28:21 eng-fxih kernel: rst= 192.168.0.1 dst= 192.168.0.2
```

Если ваш компьютер настроен на работу в сети, то вы наверняка увидите, что периодически наш интерфейс поступают и широковещательные пакеты, которые несложно узнать на однообразном фоне управляющих.

Наша следующая задача – научить драйвер принимать данные и отправлять их в сетевую подсистему ОС Linux. Для этого мы введём в него необходимые функции, а затем свяжем с аппаратным устройством, управляющим передачей данных по кабелю. В качестве такого устройства мы используем COM-порт компьютера. Таким образом мы получим программу, похожую на известный драйвер SLIP. Конечно, наш драйвер не будет поддерживать всех функций своего известного прототипа, но зато он будет намного проще и не будет требовать для работы дополнительных программ-демонов. Однако, все это будет уже в следующей части. Не пропустите! LXF

ПОЛЕЗНЫЕ ССЫЛКИ

http://linuxcenter.ru/lib/books/lkmpg.phtml – перевод «The Linux Kernel Module Programming Guide» на русский язык

КОНФИГУРАЦИЯ ЯДРА: готовые рецепты

Потерялись в make menuconfig? **Михаил Емельченков** научит, как оптимизировать ядро под ваш процессор, настроить USB и FireWire-устройства, Bluetooth, программный RAID...

Ядро – центральный компонент любой операционной системы. Ядро управляет памятью, процессами, внешними и внутренними устройствами, сетевыми протоколами и многим другим. Ядро Linux было создано Линусом Торвальдсом в 1991 году и с тех пор успешно развивается, дополняясь поддержкой нового оборудования, новых технологий и исправлениями ошибок.

На сегодняшний день Linux – монолитное ядро с поддержкой модулей. Драйверы устройств и расширения ядра могут подгружаться и выгружаться непосредственно во время работы системы без необходимости её перезагрузки.

От того, как настроено ядро, зависит функциональность системы. В этой статье мы рассмотрим, чего можно достичь с помощью некоторых опций ядра. Всё нижесказанное справедливо для последних версий ядер ветки 2.6.

Способы настройки ядра

Для настройки ядра переключимся в режим суперпользователя и перейдём в директорию, содержащую его исходные тексты:

```
$ su -
# cd /usr/src/linux
```

Существует пять способов конфигурирования ядра:

1) # make config

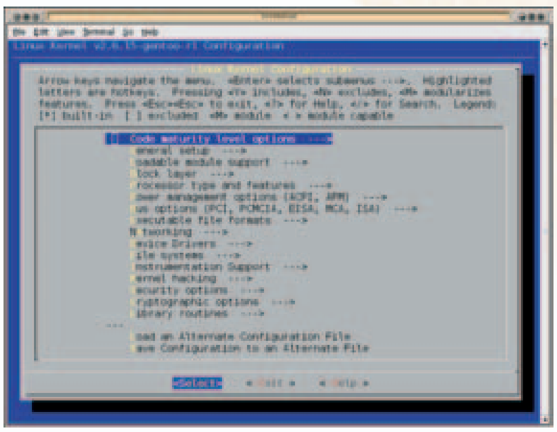
Самый примитивный и неудобный способ. Вам будут задаваться вопросы относительно настроек ядра строка за строкой, раздел за разделом. Отвечать нужно будет «y», «n» или «m», то есть «да», «нет» или «модуль».

2) # make oldconfig

В новые релизы ядра часто включаются новые опции. **make oldconfig** похож на **make config**; он будет использовать ответы из прежнего файла конфигурации, но будет задавать вам вопросы только о вновь появившихся опциях.

3) # make menuconfig

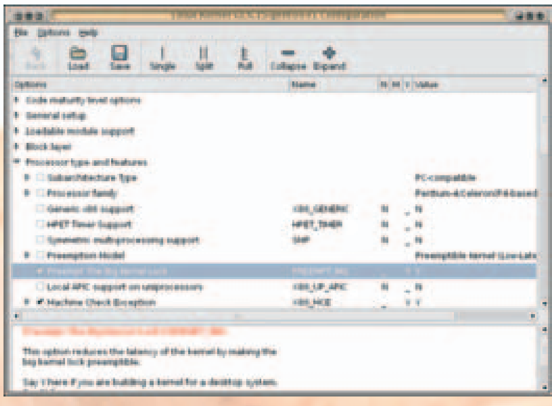
Конфигурация ядра с помощью псевдографической оболочки (интерфейс на базе *ncurses*). Наиболее широко используемый вариант при работе в консоли. Опции разделены иерархически на разделы и подразделы. Это самый универсальный способ настройки.



Make menuconfig предоставляет псевдографический интерфейс.

4) # make gconfig

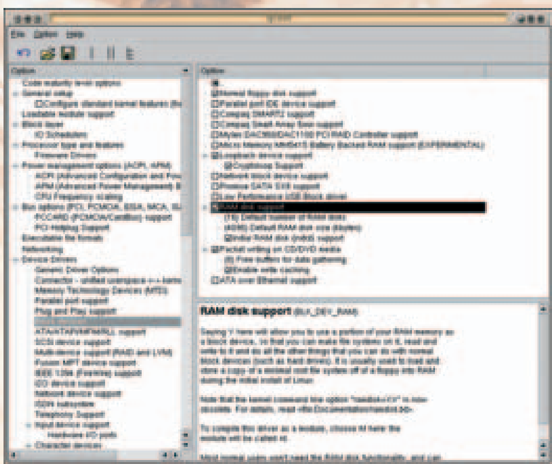
Графический интерфейс кофигурации ядра на базе библиотеки GTK+.



Make gconfig использует GTK+ и, наверное, не нравится Линусу Торвальдсу.

5) # make xconfig

Графический интерфейс конфигурации ядра на базе библиотеки Qt. Наиболее удобный из всех вышеперечисленных, т.к. отображает разделы и подразделы в виде дерева, а также показывает справку по каждому пункту.



Make xconfig собственной персоной. Как сказал Л. Торвальдс: «Просто используйте KDE».

Оптимизация

Каждый пользователь мечтает о высокопроизводительной системе, но, к сожалению, не каждый знает, как достичь высокой производительности. Ответ на этот вопрос заключается не только в покупке самого современного аппаратного обеспечения, но и в оптимизации существующего ПО. Одним из способов такой оптимизации может служить настройка ядра Linux.

В разделе Processor type and features -> Processor family выберите тип вашего процессора. Каждый из типов включает в себя специфи-

ческие инструкции, применение которых, соответственно, ускоряет работу. Но помните, что если вы неверно укажете тип процессора, ваш компьютер будет работать медленнее, если вообще загрузится. В зависимости от того, является ли ваш компьютер сервером или рабочей станцией, нужно выбрать требуемую опцию в разделе Processor type and features -> Preemption Model (модель вытеснения задач). Опция No Forced Preemption будет соответствовать серверному варианту, а Voluntary Kernel Preemption и Preemptible Kernel – настольным, причём последний позволяет работать приложениям более «гладко» даже тогда, когда система находится под высокой загрузкой. Также для более «гладкой» работы можно включить опцию Preempt The Big Kernel Lock.

В семействе процессоров Intel Pentium Pro, Pentium II и выше используются специальные регистры – MTTR (Memory Type Range Registers). Эти регистры используются для доступа процессора к различным диапазонам памяти. Включение этой опции позволит существенно повысить производительность системы, особенно если вы используете видеокарту PCI или AGP.

Про следующий, не менее важный параметр, следует вспомнить, когда вы решите увеличить оперативную память. Если у вас менее 1 Гб памяти, выбирайте опцию off в разделе Processor type and features -> High Memory Support. Владельцам 1 – 4Гб RAM подойдет 4 Gb, а самым счастливым – «64 Gb.

В случае с многопроцессорной системой (а также с системой, процессор который поддерживает Hyper Treading) следует включить опцию Symmetric Multi-Processing Support. После включения данной опции станет доступна опция Maximum number of CPUs (2 – 255). Укажите здесь реальное количество процессоров. На производительность она не влияет, но позволит сэкономить несколько килобайт на размере ядра. Также следует включить опцию Enhanced Real Time Clock Support раздела Device Drivers -> Character devices. Учтите, что в SMP-режиме не будет работать управление питанием «Advanced Power Management (APM)».

Программный RAID

Чем программный RAID лучше аппаратного? Аппаратный RAID, как правило, строится на недешёвых контроллерах и SCSI-дисках, дешёвые контроллеры для дисков IDE (и SATA) часто оказываются псевдоаппаратными контроллерами, т.е. большинство их функций перекладывается на Windows-драйвер. Поэтому в случае невозможности использования аппаратного контроллера можно организовать программный RAID.

Для его поддержки необходимо включить опции ядра в разделе Device Drivers -> Multi-device support (RAID and LVM) -> Multiple devices driver support (RAID and LVM) -> RAID support. Здесь нужно выбрать опции в зависимости от потребностей (RAID-0, RAID-1, RAID-10 и т.д.). Одна из наиболее популярных утилит для работы с RAID называется mdadm. Более подробно о ней, а также вообще о программном RAID, можно узнать на **http://www.tldp.org/HOWTO/Software-RAID-HOWTO.html**.

Драйверы аппаратных RAID-контроллеров находятся в разделах Device Drivers -> Block devices, Device Drivers -> ATA/ATAPI/MFM/RLL support, Device Drivers -> SCSI device support -> SCSI low-level drivers, Device Drivers -> SCSI device support -> SCSI low-level drivers -> Serial ATA (SATA) support.

USB

За поддержку USB отвечает раздел «Device Drivers» -> «USB support». В подразделе «Support for Host-side USB» производится основная конфигурация USB, обычно тут следует выбирать «EHCI HCD (USB 2.0) support» для поддержки USB 2.0 и «UHCI HCD (most Intel and VIA) support» для поддержки USB 1.1.

Поддержка USB-принтеров включается через «USB Printer support». После компилирования ядра с этой опцией и установки драйверов принтера (при необходимости) с принтером можно работать как обычно, например, с помощью CUPS.

Особый интерес представляют так называемые «USB Storage» – всевозможные накопители, подключаемые через USB. Одной из наиболее полезных разновидностей таких накопителей являются флеш-карты, например, карты SD, используемые в фотоаппаратах, плейерах, принтерах и прочей микропроцессорной технике. Считывание и записывание данных с таких карт происходит с помощью адаптера карт памяти (кард-ридера).

Для включения поддержки «USB Storage» отметьте опцию SCSI disk support раздела Device Drivers -> SCSI device support, а также опцию USB Mass Storage support из раздела Device Drivers -> USB support. Если у вас какой-то специфичный кард-ридер, здесь можно найти набор подопций для конкретных моделей, но в большинстве случаев должно работать и так.

Если с опцией USB Mass Storage support всё понятно, то необходимость включения SCSI disk support требует некоторой аргументации. Так вот, это делается потому, что взаимодействие с USB-накопителем происходит посредством эмуляции SCSI, а соответствующие дисковые разделы имеют названия **/dev/sda1, /dev/sdb1** и т.д.

Работа с устройствами происходит обычным образом. Сначала создаётся точка монтирования, затем монтируется само устройство:

```
# mkdir /mnt/flash
# mount /dev/sda1 /mnt/flash
# ls /mnt/flash
# umount /mnt/flash
```

Также существует альтернативный драйвер для USB-накопителей. Называется он Low Performance USB Block driver и находится в разделе Device Drivers -> Block devices. При его использовании разделы именуются не **/dev/sda1**, **/dev/sdb1** и т. д., **а /dev/uba1**, **/dev/ubb1** и т. д. Это низкопроизводительный драйвер. Он медленнее «USB Mass Storage support», но более надёжный. Этот драйвер работает в обход эмуляции SCSI. Благодаря тому, что он реализует только самые необходимые для работы функции, с его помощью можно заставить работать более широкий спектр устройств. Поэтому, если ваше устройство не работает с «USB Mass Storage support», попробуйте этот драйвер.

FireWire

FireWire (она же IEEE 1394, она же i.Link) – шина, по которой происходит обмен данными внешних устройств и компьютера со скоростью до 400 Мбит/с. Наболее распространённое её применение – это передача видео с цифровой (DV) видеокамеры и организация сети Ethernet.

Поддержка этой шины включается в разделе Device Drivers -> IEEE 1394 (FireWire) support -> IEEE 1394 (FireWire) support. Большинство карт соответствует спецификации OHCI-1394, поэтому включите опцию «OHCI-1394 support».

Для работы с DV-камерами нужно также включить опции «OHCI-1394 Video support», «OHCI-1394 I/O support» и «Raw IEEE1394 I/O support». После этого можно будет воспользоваться такими программами для захвата и редактирования видео, как Kino (**http://www.kinodv.org**), Cinelerra (**http://heroinewarrior.com/cinelerra.php3**), MainActor (**http://www.mainconcept.com/mainactor.shtml**) и другими. Не забудьте сделать деинтерлейсинг изображения, полученного с DV-камеры (что такое деинтерлейсинг, можно прочитать в энциклопедии – **http://en.wikipedia.org/wiki/Deinterlacing**).

С помощью FireWire можно также организовать сеть Ethernet. Для этого нужно включить опцию Ethernet over 1394. В таком случае можно будет соединить две Linux-станции посредством FireWire. Для работы с системами Windows XP и Mac OS X необходимо также включить опцию Build in extra config rom entries for certain functionality -> IP-1394 Entry (NEW)». По стандарту длина кабеля не должна превышать 4.5 м. С помощью удлинителей можно довести длину кабеля до 10 м, но это уже нестандартное решение, поэтому с ним возможны проблемы. Хотя FireWire и выигрывает в скорости у стандарта 100BaseT (100 Мбит/с, длина кабеля до 100 метров), он проигрывает в расстоянии, а с появлением Gigabit Ethernet соединение посредством FireWire проигрывает во всём. Одним из наиболее оправданных случаев применения Ethernet-сети на базе FireWire является соединение ноутбука с настольным компьютером, т.к. современные ноутбуки обычно комплектуются портом FireWire.

Работа с сотовым телефоном через Bluetooth

Bluetooth – спецификация, которая позволяет использовать вместе такие устройства, как компьютер, КПК или телефон. Связь осуществляется на короткие расстояния (10 м, 100 м) по радиоканалу.

Большинство Bluetooth-адаптеров подключаются по USB, поэтому удостоверьтесь, что вы собрали ядро с поддержкой USB.

Поддержка Bluetooth включается в разделе Networking -> Bluetooth subsystem support. В этом разделе следует включить опцию L2CAP protocol support, требуемую для работы большинства Bluetooth-приложений. Также потребуется SCO links support, если вы планируете использовать Bluetooth для передачи голоса, например, через гарнитуру. Опции RFCOMM protocol support и RFCOMM TTY support нужны для поддержки таких способов применения Bluetooth, как Dialup и OBEX (OBEX – OBject EXchange – протокол для передачи объектов: контактов, файлов и т.д.). Включение BNEP protocol support понадобится для поддержки так называемых Bluetooth PAN (Personal Area Network) для объединения нескольких Bluetooth-адаптеров в сеть. Ну и ещё одна опция – HIDP protocol support – требуется для поддержки HID (Human Interface Devices), то есть устройств, которые предназначены для управления компьютером, таких как клавиатура, мышь и т.д.

Перейдем к подразделу Bluetooth device drivers. Наиболее распространёнными являются USB-адаптеры, поэтому выбираем опцию HCI USB driver и подопцию SCO (voice) support. Если же у вас адаптер с интерфейсом, отличным от USB, выбирайте одну из подходящих опций данного подраздела. Не забывайте, что для поддержки USB-устройств должны быть активированы соответствующие опции в разделе Device Drivers -> USB support.

Теперь ядро поддерживает Bluetooth. Но для работы приложений необходимо поставить официальный стек Bluetooth – BlueZ (**http://www.bluez.org**). Соответствующие пакеты в вашем дистрибутиве могут называться **bluez-libs**, **bluez-utils** или подобным образом.

Для работы с файлами на сотовом телефоне через Bluetooth в консоли можно воспользоваться программой ObexFTP (**http://triq.net/obex**). Для работы с сотовым телефоном в графической среде – GNOME Phone Manager (**http://usefulinc.com/software/phonemgr/**), KMobiletools (**http://kmobiletools.berlios.de**) и другими.

NFS и SMB

Network File System (NFS) в мире Linux – почти то же самое, что папки общего доступа Windows (по крайней мере, по области применения). Главное отличие состоит в том, что соединяются не Windows-станции, а компьютеры Unix.

Для включения поддержки NFS в ядре отметьте опции NFS file system support, Provide NFSv3 client support, NFS server support, Provide NFSv3 server support в разделе File systems -> Network File Systems.

Для работы с NFS применяются стандартные утилиты. Обычно они содержатся в пакете nfs-utils или подобном. Для работы NFS также необходимо установить и запустить сервис portmap. Теперь можно монтировать директории командой вида:

```
# mount x.x.x.x:/directory /mount_directory
```

Где **«x.x.x.x»** – IP-адрес или DNS-имя NFS-сервера, **«/directory»** – экспортированная директория, которую вы хотите подключить, **«/mount_directory»** – локальная точка монтирования для экспортируемой директории.

При наличии в системе демона **nfsmount**, можно также монтировать сетевые директории при загрузке. Для этого следует добавить их в **/etc/fstab**:

```
x.x.x.x:/directory/mount_directory    nfs    rw 0    0
```

Для работы с NFS в качестве сервера, отредактируйте файл **/etc/exports**:

```
/directory x.x.x.x/y.y.y.y(async,no_subtree_check,rw)
```

В данном случае директория будет доступна всей подсети **«x.x.x.x»** маски **«y.y.y.y»**. Более подробно о способах задания экспортируемых директорий читайте в **«man exports»**.

Для работы в качестве NFS-сервера не забудьте запустить демон **nfs**.

SMB (Server Message Block) – протокол, предназначенный для обеспечения общего доступа к файлам и принтерам в локальных сетях. Для включения поддержки SMB в ядре отметьте опцию SMB file system support (to mount Windows shares etc.) раздела File systems -> Network File Systems. Также отметьте опцию CIFS support (advanced network filesystem for Samba, Window and other CIFS compliant servers) того же раздела, если вы планируете подключаться к серверам на базе Windows 2000, Windows 2003, Windows XP и некоторым другим.

Далее необходимо установить программы для работы с SMB. Обычно этот пакет называют коротко – samba.

В средах KDE, Gnome, XFCE пользоваться SMB можно с помощью штатных средств, но существует также возможность работы с SMB в консоли. Для монтирования общедоступной папки служит команда **smbmount**:

```
# smbmount //<windows machine name>/<shared folder> /<mountpoint>
-o username=<user>,password=<pass>,uid=1000,umask=000
```

Здесь <**windows machine name**> – имя файл-сервера, <**shared folder**> – папка на сервере, <**mountpoint**> – точка монтирования, <**user**> и <**pass**> – имя пользователя и пароль соответственно.

Отмонтирование выполняется командой **smbumount**:

```
# smbumount <mountpoint>
```

Посмотреть список общедоступных папок можно с помощью команды

```
# smbclient -L <windows machine name> -U%
```

Также полезна утилита smbtree. Она выполняет функции «Сетевого окружения» Windows, то есть показывает дерево всех доменов, серверов в этих доменах и их сетевых ресурсов.

Конфигурация сервера выполняется с помощью файла **/etc/samba/smb.conf**. По умолчанию он пустой, но за основу можно взять файл **/etc/samba/smb.conf.example**. В этом файле даны примеры настройки общедоступных папок и принтеров.

Кроме NFS и SMB существует поддержка ещё нескольких сетевых файловых систем, таких, как NCP (для монтирования томов NetWare), Coda (сетевая файловая система, похожая на NFS, но обладающая более широкими возможностями), экспериментальная поддержка AFS и Plan 9 Resource Sharing.

TUN/TAP

Драйвер TUN/TAP используется для создания туннельных интерфейсов. Иными словами, с помощью него и сопутствующих программ можно организовать простейшую виртуальную частную сеть, то есть VPN-соединение.

Включается драйвер опцией Universal TUN/TAP device driver support раздела Device Drivers -> Network device support.

VPN-соединение организуется с помощью таких программ, как VTun (**http://vtun.sourceforge.net/**), OpenVPN (**http://openvpn.net/**).

Драйвер TUN/TAP также используется для организации сети между виртуальной машиной и рабочей средой в таких эмуляторах ПК, как Bochs (**http://bochs.sourceforge.net/**) и QEMU (**http://fabrice.bellard.free.fr/qemu/**).

Шифрование

В ядре Linux есть раздел Cryptographic options. Это реализация Cryptographic API – стандартного интерфейса для шифрования, хеширования и даже сжатия данных. Пользователю предоставляется на выбор множество алгоритмов: здесь есть и SHA256 digest alorithm, и AES cipher algorithms, и многие другие.

Cryptographic API используется модулями ядра, например, драйверами Wi-Fi карт, а также для поддержки шифрования с помощью device-mapper.

Для шифрования дисков с помощью device-mapper можно использовать программу *Dm-crypt* (**http://www.saout.de/misc/dm-crypt/**).

Cryptographic API также используется для шифрования сетевого трафика, например, в *IPSec*. IPSec-туннель можно реализовать с помощью *Openswan* (**http://www.openswan.org**). LXF

Пишите нам

Ядро Linux неисчерпаемо, как и атом. Какие из интересных вам тем остались за рамками этой статьи? Пишите нам: **letters@linuxformat.ru** и, возможно, мы продолжим этот разговор на страницах журнала.

Учебники >>

Наши эксперты помогут вам с любым приложением Linux

БОГАТЕНЬКИЙ БУРАТИНО

Евгений Балдин считает: «Помощь сообществу начинается с помощи себе»

Прошла информация, что чистая прибыль Mozilla Corporation за последний год составила около пятидесяти (!) миллионов долларов. Mozilla Corporation была основана в 2005 году некоммерческой организации Mozilla Foundation. По замыслу, подобная структура позволяет заниматься бизнесом (и, как следствие, платить налоги). Mozilla Foundation является единственным собственником Mozilla Corporation. Это, по идее, даёт гарантию что деньги пойдут на развитие Mozilla, а не осядут в чьих-то карманах. Посмотрим, как это будет выглядеть на практике.

Для меня это был пример того, что открытые разработки позволяют зарабатывать и зарабатывать неплохо, даже чисто-программистским проектам. Это было очевидно. Но когда я увидел комментарии на LORe (www.linux.org.ru) и ещё на паре «околокомпьютерных» сайтах, то у меня появились сомнения в адекватности человечества (как минимум его части). Вместо того, чтобы порадоваться за ребят, участники дискуссии призывали чуть ли не линчевать сотрудников корпорации. И вправду, зависть – страшная сила.

Свободные продукты – не значит бесплатные. За свободу необходимо платить. Потому что зарабатывать «простому человеку» в условиях свободы проще.

info@linuxformat.ru

КОД В ЭТОМ РАЗДЕЛЕ

Мы надеемся, что некоторые простые правила позволят избежать недоразумений. Если строка кода оказывается слишком длинной, она переносится на следующие, причем все они обрамляются синим прямоугольником:

```
procedure
TfrmTextEditor.mniWordWrapClick
(Sender: TObject);
```

В противном случае, между строками будет небольшой промежуток:

```
mniWordWrap.Checked := false
end;
```

Как правило, весь код можно найти на прилагаемом к журналу диске.

В ЭТОМ МЕСЯЦЕ ИЗУЧАЕМ...

76 ПЕРВЫЕ ШАГИ
Evolution: горькая правда, сладкая ложь или прекрасный органайзер? Спросим у **Энди Ченнела**!

80 GIMP
Хорошей программе – хорошую иконку! И логотип... **Майкл Дж. Хэммел** поможет вам – вы поможете Linux на рабочем столе.

84 INKSCAPE
Дмитрий Кирсанов научит не только рисовать контуры, но и набирать текст

88 HARDCORE LINUX
Присоединяйтесь к закрытому клубу лиц, который знают, зачем на самом деле нужен grep. **Пол Хадсон** поручился за вас.

92 PHP
Пол Хадсон делает все, чтобы вы не вытащили секреты из его БД

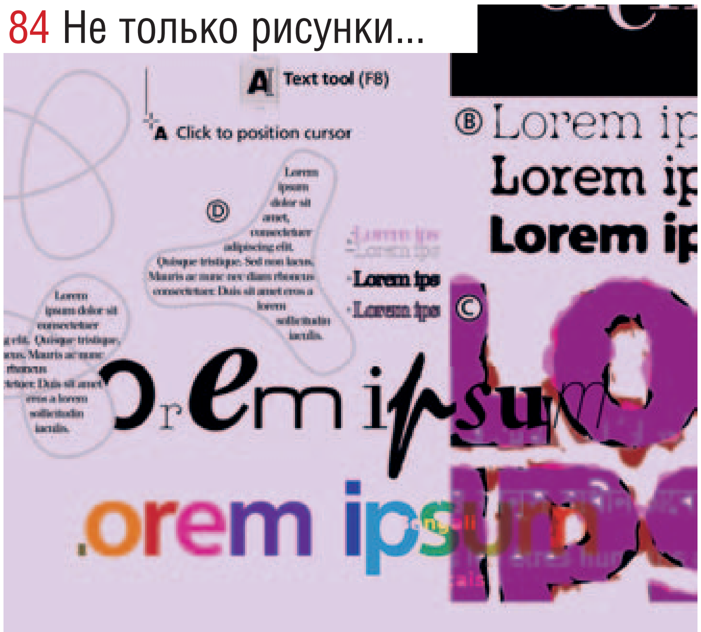
94 QT: ПЕРВЫЕ ШАГИ НОВАЯ СЕРИЯ
Андрей Боровский открывает для вас мир графических приложений Linux. И не только...

98 ПОЧЕМУ – VIM?
Владимир Попов высказывается в поддержку любимого редактора

102 УЧЕБНИК PYTHON, ЧАСТЬ 4
Объектно-ориентированный **Сергей Супрунов** – ярчайший экземпляр класса программистов на Python

104 METAPOST, ЧАСТЬ 3
И.. раз! И.. два! **Евгений Балдин** умеет командовать компьютерами

84 Не только рисунки...



СОВЕТ МЕСЯЦА ДВЕ BASHни

Приглашение *Bash* – одна из неизменных вещей, которую большая часть людей воспринимает, как данность. Это странно, поскольку изменить его текст очень легко. В большинстве дистрибутивов используется **приглашение вида** <пользователь@домен>, например 'user@localhost'.

Ключ к решению – динамические переменные, которые позволяют внедрить в приглашение нужную информацию. Чтобы создать приглашение типа <пользователь@домен>, наберите и выполните или добавьте в ваш **~/.bash_profile** или **~/.bashrc** такую строку:

```
export PS1="[\u@\h]$"
```

Команда **export** устанавливает переменную окружения PS1. Она содержит текст приглашения, который и указывается в двойных кавычках после знака равенства.

Его ценность заключается в динамических переменных, расположенных по обе стороны от символа '@'; **\u** – имя пользователя, **\h** – первая часть имени хоста. Вы можете использовать и другие, например, **\w** (текущий рабочий каталог), **\d** (текущая дата) или **\!** (текущая позиция в буфере команд). Если эти переменные кажутся вам знакомыми, то это потому, что они являются escape-последовательностями, применяемыми во многих языках программирования.

Существуют и другие виды escape-последовательностей. Эта, например, меняет имя окна консоли:

```
"\e]2;title\a"
```

В свое время хакеры соревновались в сложности своих приглашений Bash. Диапазон возможных вариантов действительно не ограничен. Используя условный оператор **if**, можно менять текст приглашения в зависимости от номера сессии, однако, с помощью escape-последовательностей можно просто добавить в него полезную информацию.

ПЕРВЫЕ ШАГИ СЕРИЯ «LINUX ДЛЯ НОВИЧКОВ»

Evolution Управляйте своей почтой

НА ДИСКЕ

Если вам нужно поддерживать порядок в почтовом ящике, вести записную книжку или управлять проектами, то это несложное руководство по секретам Evolution сослужит вам хорошую службу. Ваш тренер **Энди Ченнел** уже готов вас натаскать…

МЕСЯЦ НАЗАД

Мы прогулялись по блогам, познакомились с новым поколением сетевых сервисов и такими программами как Flock, Flickr и Digikam.

Возможно, я один такой, но мой объем работы и отведенное на него время всегда находятся в обратной зависимости. Я думаю, именно поэтому люди планируют построить дом за три недели, а в итоге возятся три года (превышая задуманный бюджет в 330 раз). Подобным проектам, как правило, не хватает грамотного планирования. Именно об этом мы и поговорим в этот раз, а главным приложением для работы будет *Evolution*.

Большинство людей считают, что *Evolution* – это высего лишь хороший e-mail-клиент, однако программа способна на гораздо большее. Ее ключевая особенность состоит в способности объединять огромные объемы информации, причем пользователь может создавать ссылки между различными фрагментами. Например, вы вносите в органайзер *Evolution* какое-то важное задание, которое нужно успеть сделать к определенному сроку. Можно сделать так, чтобы программа автоматически предупреждала всех связанных с этим заданием людей за несколько дней до крайнего срока, без вмешательства забывчивых участников. Видите, какое удобное планирование? В этой статье я расскажу вам, как создать в *Evolution* группу контактов, собрать все относящиеся к ней письма, организовать коллективную работу и многое другое с помощью сетевого органайзера. Не думайте, что все эти возможности нужны лишь профессионалам или занятым время от времени людям – конечно, они подходят для управления деловыми проектами, но и все те, кто получает очень много электронной почты и переживает за каждое письмо, извлекут из *Evolution* большую пользу.

> «ЭТОТ ФУНКЦИОНАЛ НУЖЕН НЕ ТОЛЬКО ПРОФИ ИЛИ НЕПОСТОЯННЫМ РАЗРАБОТЧИКАМ LINUX»

Чтобы не терять времени, давайте договоримся, что *Evolution* у вас установлен (программа есть на нашем диске) и настроен на ваш почтовый ящик. Если это еще не сделано, вам поможет великолепный мастер первого запуска, который проведет вас через все этапы настройки.

Создаем группу

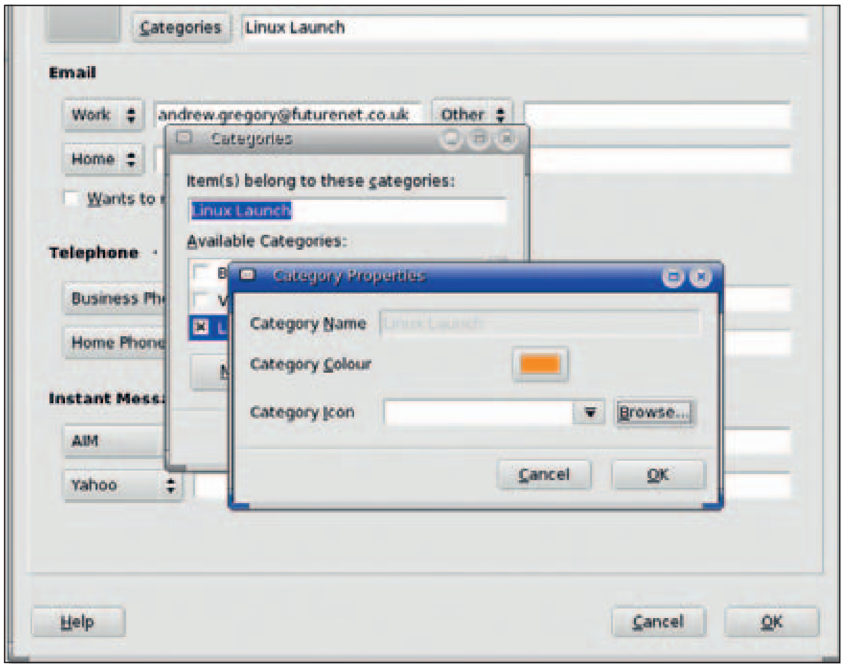
Первым шагом на пути к грамотному планированию будет определение состава вашей команды. Это могут быть сотрудники отдела, участники научного исследования, или даже ваша семья (вдруг вы планируете двухнедельный отдых на Канарах?) – какой бы ни была ваша цель, с помощью *Evolution* вы сможете объединить людей вокруг нее.

Для примера я создам небольшую группу из четырех человек (я и еще трое). Будем считать, что от каждого коллеги я получил по электронному письму, а значит, часть информации у меня уже есть. Остальное я внесу в *Evolution* самостоятельно. Я могу организовать несколько упорядоченных адресных книг. Проще всего это сделать, щелкнув по очереди правой кнопкой мыши по именам людей в графе «От:» («From:») и

выбрав пункт «Добавить в адресную книгу» («Add to address book»). Вы можете просто внести имя в адресную книгу, или же перейти к более подробному экрану. Выберите последнее и внесите необходимую информацию для записи. Именно с этого момента возможности *Evolution* выходят за рамки обычного органайзера.

По умолчанию, в *Evolution* создается одна адресная книга на каждую запись, но мы создадим вторую. В левой части окна (в режиме «Контакты» - «Contacts»), щелкнете правой кнопкой мыши под надписью «На этом компьютере» («On this computer») и выберите пункт «Новая адресная книга» («New address book»). В качестве типа оставьте значение «На этом компьютере», а имя можете задать любое. Теперь разверните снова Редактор контактов (он ведет себя как самостоятельное окно в панели задач) и выберите в поле «Где:» («Where:») только что созданную адресную книгу.

Кое с чем мы уже разобрались, но если вы имеете дело с большим числом людей, вам захочется как-то упорядочить их персональную информацию. В таком случае, вам пригодится возможность сортировки контактов по категориям (см. изображение справа). По умолчанию в списке содержится набор стандартных категорий типа «День рождения» и «Праздник», но вы также можете создать свои категории. Нажмите кнопку «Категории», затем «Создать», и придумайте имя. По желанию можно также присвоить категории цвет или определенную иконку-значок (я выбрал оранжевый цвет), которые будут подсвечивать все заметки и события, связанные с группой контактов. Создание групп и категорий помогает в дальнейшем находить нужные данные, поэтому не поленитесь сделать это заранее.



Сортировка данных по категориям помогает в дальнейшем находить то, что нужно.

Сортируем почту

Теперь отсортируем входящую почту и пометим все письма, которые имеют отношение к нашему проекту. Для этого в *Evolution* есть два способа, причем оба основаны на принципе фильтра. Таким образом, выбрав критерий отбора, вы получите только те письма, которые пройдут через фильтр, а значит, будут относиться только к вашему проекту.

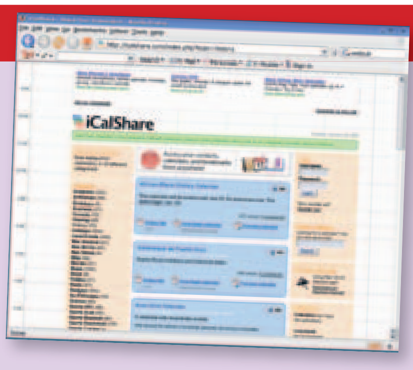
Первый способ сортировки подразумевает создание новой почтовой папки. В режиме «Почта» щелкните правой кнопкой мыши по панели «На этом компьютере», выберите «Создать» и введите имя. После этого нужно настроить программу так, чтобы входящие сообщения попадали в эту папку. Перейдите в окно настроек фильтров («Сервис» > «Фильтры») и выберите «Добавить». Если у вас установлена самая новая версия *Evolution* (2.6), то эту опцию вы также найдете в меню «Правка» > «Фильтры сообщений». После того, как фильтру дано имя, мы должны определить, что именно должно через него проходить. Настройка фильтра заключается в создании ряда условий типа «Если» – «То», которые оформлены в виде выпадающих списков.

У всех участников нашего проекта электронный адрес заканчивается на «futurenet.co.uk», поэтому давайте внесем это условие: выбрав в первом списке «Отправитель», а во втором – «Содержит», дадим значение «futurenet.co.uk». Однако я хочу, чтобы в мою папку попадали письма не от всего издательства, а только от *Linux Format*. Зная, что у каждого сотрудника редакции в подписи сообщения автоматически ставится название журнала, я могу, к примеру, исключить из входящих писем любого человека из Mac Format. Для этого я создам второе условие: «Тело сообщения» – «Не содержит» – «Mac Format». Таким образом мы создали фильтр на основе критерия отбора.

Второй раздел называется «То»; он определяет действие фильтра в заданных условиях. Мы можем выбрать любое действие из предложенного списка, например, переместить все отфильтрованные письма в специальную папку (может пригодиться, так как они вообще не попадут во «Входящие»), или выбрать что-то менее радикальное, к примеру, изменение цвета текста. Поскольку ранее мы уже назначили оранжевый цвет для группы контактов, то теперь есть смысл сделать то же самое

>>

СОВЕТЫ



- По моему опыту, даже небольшие проекты имеют свойство собирать вокруг себя множество файлов, поэтому имеет смысл сделать поддиректорию для хранения дополнительных материалов. Это не только упростит создание резервных копий данных но и сэкономит те 35 минут, в течение которых вы будете искать нужный файл на рабочем столе (вы точно знаете, что он там...)
- Используйте календарь *Evolution* для регулярного создания резервных копий важных документов. Создайте события для набросков, черновиков и прочих стадий создания документа с тем, чтобы потом перетащить файлы из файлового менеджера на панель вложений в соответствующие категории. У вас будет полный отчет о создании каждого документа. Не забывайте, однако, что при удалении события, все связанные с ним файлы также уничтожаются, и вернуть их невозможно!
- Добавьте больше столбцов в разделе задач. Щелкните правой кнопкой мыши по заголовку имеющегося столбца и выберите «Добавить столбец...» («Add a column»). Неотъемлемыми параметрами являются «Срок выполнения» («Due date») и «%Завершено» («%Complete»). Последний пункт настраивается при помощи выпадающего меню, поэтому вы можете давать задачам разные приоритеты.
- Подключитесь к удаленным календарям, воспользовавшись сетевой функцией программы. Создайте новый календарь, указав в качестве типа «В Сети» («On the web») и добавьте URL. В интернете существует большая бесплатная коллекция календарей для любых нужд: **www.icalshare.com**.
- Создайте собственные сетевые календари. Внеся необходимую информацию (публичные мероприятия или что-то личное), щелкните правой кнопкой мыши по календарю и выберите «Save to disk». Календарь будет сохранен в формате .ics (Internet Calendaring and Scheduling), и вы сможете отослать его на сайты типа **www.icalshare.com** или отправить друзьям, у которых установлено совместимое с iCalendar приложение. Вы всегда будете держать руку на пульсе событий.

для писем от всех адресатов в этой группе. Вы даже можете заставить программу издавать разные звуки при срабатывании разных критериев – совсем как персональные мелодии в телефоне.

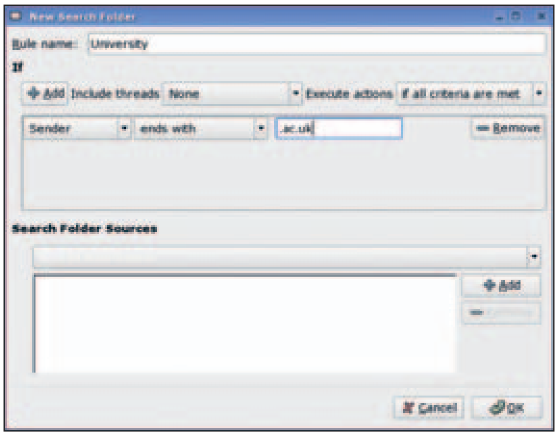
Каким бы ни был фильтр, по умолчанию он начинает действовать для новых писем. Тем не менее, вы можете применить фильтр для текущего содержания папки «Входящие». Выделите все письма (Ctrl+A) и выберите пункт «Действия» > «Применить фильтры» или нажмите Ctrl+Y.

Кроме фильтров, в *Evolution* имеются «умные» виртуальные папки. В старых версиях программы они назывались vFolders, а начиная с версии 2.4 – Search Folders. Вне зависимости от названия, эти папки

«ВИРТУАЛЬНЫЕ ПАПКИ ПОЗВОЛЯЮТ СОРТИРОВАТЬ ПОЧТУ, НЕ ПЕРЕМЕЩАЯ ЕЕ ИЗ ПАПКИ «ВХОДЯЩИЕ».»

позволяют вам сортировать почту по категориям, но в отличие от первого способа, письма не перемещаются из папки «Входящие». Search Folders работает следующим образом: если входящее письмо удовлетворяют заданному критерию, то на него создается ссылка, которая помещается в виртуальную папку. Изюминка заключается в том, что критерии можно менять «на лету» сколь угодно много раз, причем физическое расположение письма (к e-mail это относится весьма условно) при этом не меняется.

Чтобы создать новую виртуальную папку, щелкните правой кнопкой мыши по надписи Search Folders в левой части окна и выберите

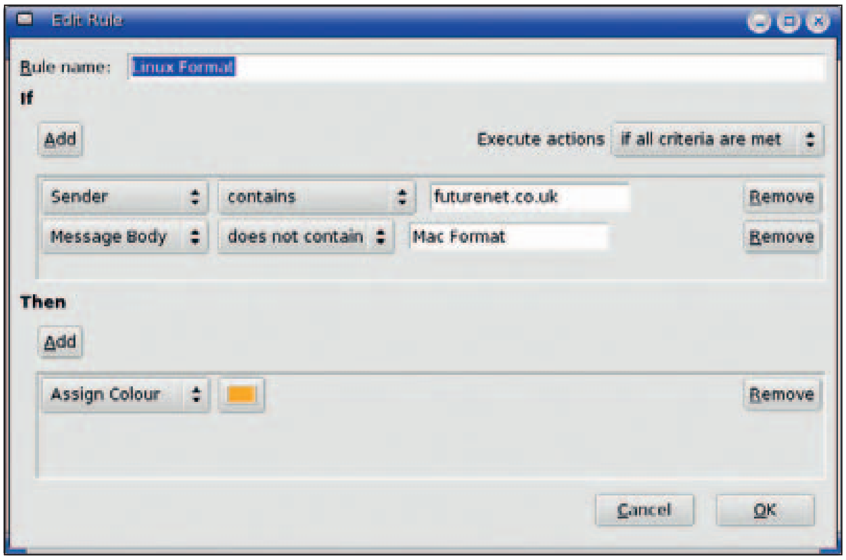


Виртуальные папки помогут собрать данные из разных источников и поместить их в одно место.

«Создать папку». Появится диалоговое окно, очень похожее на окно настройки фильтра, только без раздела «Тогда». Вместо этого, на все входящие письма, которые удовлетворяют критериям «Если», будут созданы ссылки в виртуальной папке. Наверно, это самый лучший способ хранить критерии поиска. Чтобы изменить настройки виртуальной папки, зайдите в ее свойства правым щелчком мыши. Замечательная особенность этого метода сортировки состоит в том, что Search Folders реагируют на письма точно так же, как и обычная папка «Входящие», поэтому, если вам пришло письмо с пометкой «Важно», то виртуальная папка (настроенная на фильтрацию именно таких сообщений) выделится жирным, и вы сразу поймете, что на письмо следует обратить внимание.

Другая полезная функция *Evolution*, о которой не часто вспоминают, называется «Пометить к исполнению («Mark as follow up»). Ее суть заключается в том, что помимо даты и времени, к письму можно прикрепить напоминание о каком-либо важном сроке. Щелкните правой кнопкой мыши по письму и выберите этот пункт, установив нужное время и число месяца. Эта информация будет видна над свойствами письма, когда вы щелкнете по нему в следующий раз. Также в этом разделе вы можете прикрепить к письму и другие флаги: «Не пересылать», «Позвонить», «Можно не отвечать» и прочие. Схожим образом вы можете превратить письмо в задачу, щелкнув по нему правой кнопкой мыши. Новая задача будет добавлена в соответствующий раздел, где ей можно назначить дату и время напоминания. Это очень удобно для заблаговременного оповещения о приближении крайнего срока.

Все эти инструменты помогают эффективно управлять проектом, но теперь, когда мы разобрались с созданием контактов и управлением письмами, пора взглянуть на область, где эти элементы собираются вместе: календарь.



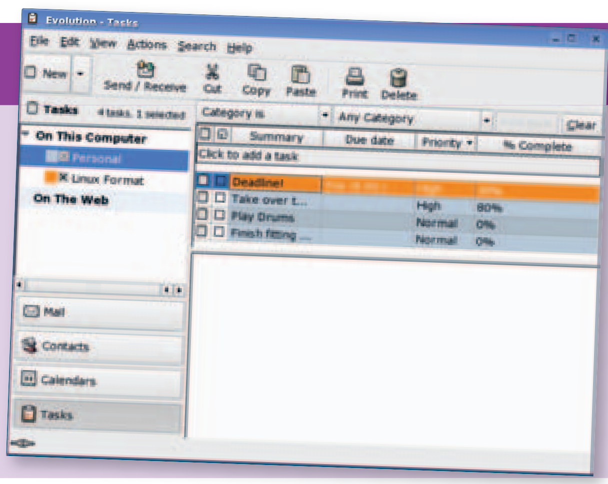
Фильтры *Evolution*, основанные на условиях, невероятно гибки в настройке.

ИСПОЛЬЗУЕМ МЕНЕДЖЕР ЗАДАЧ

Когда вы начинаете запутываться... Напишите план!

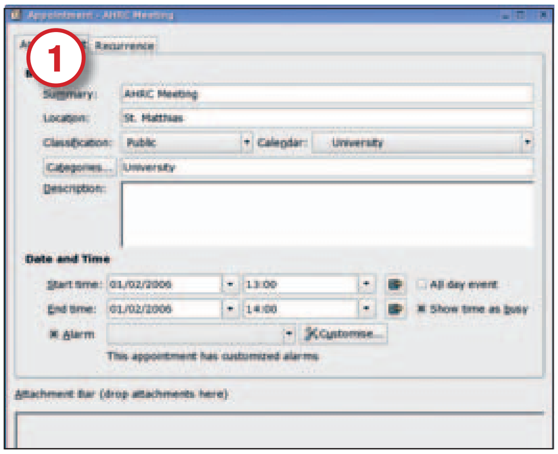
При подготовке к какому-либо событию иногда есть смысл разбить задачи на более управляемые разделы. Технология, которую использует *Evolution*, существует еще с древнего Вавилона: это план!

Нам понадобится Менеджер Задач, который представляет собой удобный инструмент для создания списков. Мы записываем задачу, указываем дату ее начала и завершения, а также статус, определяющий ее важность. Задачи группируются в список и могут быть отсортированы по любому параметру: важности или датам. Подобно тому, как мы поступали с контактами, здесь можно группировать задачи по цвету. Еще можно дать задание другому человеку (из списка контактов) и автоматически переслать ему все детали по почте. Для этого щелкнете правой кнопкой мыши по заданию, выберите «Назначить задание» («Assign task») и введите необходимые данные.



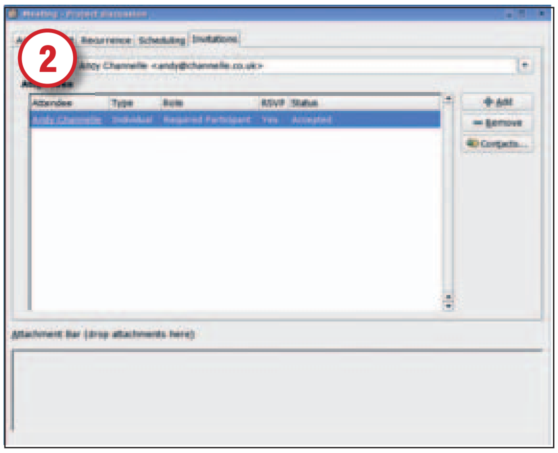
КАК ПОЛЬЗОВАТЬСЯ КАЛЕНДАРЕМ В EVOLUTION

***Evolution* способен работать с несколькими календарями** одновременно. Вы можете «включать» или «отключать» их в левой части экрана в разделе «Календари». Ранее мы задали определенный цвет для всех участников проекта, а теперь сможем присвоить его нашему проектному календарю. Создайте календарь, выбрав пункт «Файл» > «Создать» > «Календарь» или нажмите Ctrl+Shift+N. Дайте календарю имя и цвет, желательно тот же, что мы использовали ранее.



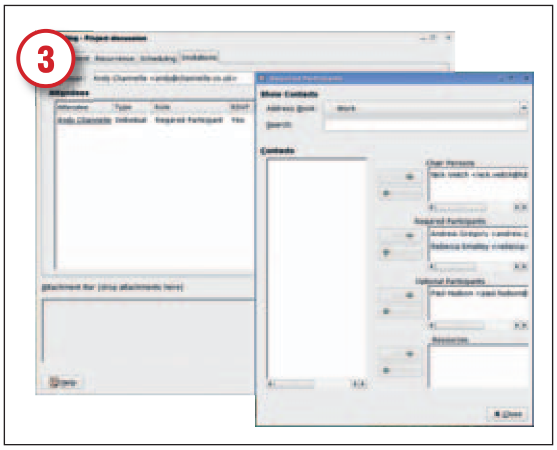
Добавляем событие

Самый простой способ начать работу с календарем заключается в щелчке правой кнопкой мыши по дню/дате и выборе пункта «Новая встреча» («New appointment»). Вы сможете ввести информацию о событии и включить напоминание, которое оповестит вас о нем заранее. Вы можете выбрать тип напоминания (всплывающее окно, звук, предупредительное письмо), а также установить время его срабатывания. Запланированные события выделяются в календаре выбранным вами цветом, и их легко заметить. Это все прекрасно, однако хороший менеджмент не ограничивается лишь вашим расписанием, а касается еще и личного времени ваших коллег.



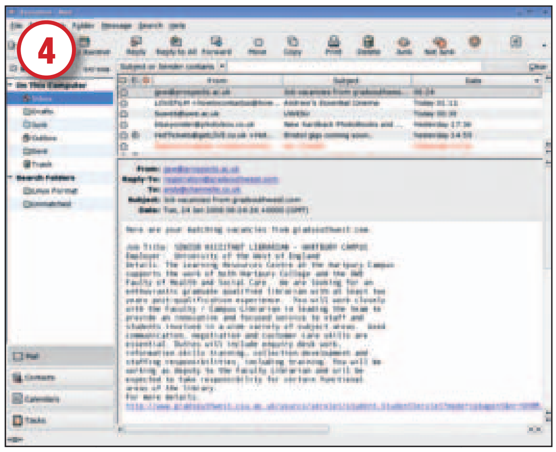
Планируем общее собрание

Следующим шагом будет планирование встречи с коллегами при помощи *Evolution*. Щелкнете правой кнопкой мыши по нужной дате и выберите «Новое собрание» («New meeting»). Диалоговое окно похоже на таковое в настройках событий, оно содержит поля для настройки даты, времени и оповещений о собрании. Также здесь есть возможность пригласить на собрание ваших коллег – перейдите на вкладку «Приглашения», где вы автоматически становитесь организатором собрания. Каждое мероприятие может иметь только одного организатора. Главная часть окна содержит список посетителей, и именно здесь (как легко догадаться) мы определим, кто именно должен присутствовать на собрании.



Определяем статус участников

Нажав на кнопку «Контакты», мы можем добавить любых участников из адресной книги. Здесь же, нажатием на соответствующие кнопки, определяется и роль каждого участника собрания. Нажмите OK для публикации данных о встрече. Кроме определения ролей участников, здесь также можно изменять статус приглашения: принято, отклонено или что-то другое.



Рассылаем приглашения

Последним штрихом будет сбор необходимых документов и иллюстраций в панель вложений в нижней части окна (Attachment bar) и их рассылка всем участникам. Приглашения отсылаются в виде вложений *iCal*, поэтому при получении почты они автоматически добавятся в календари участников (при условии, если они пользуются *Evolution* или другим iCal-совместимым приложением), а вы получите уведомление о доставке. Изменения, внесенные вами в мероприятия, автоматически рассылаются всем участникам, так что они всегда будут в курсе дела.. LXF

ЧЕРЕЗ МЕСЯЦ

Мы займемся обработкой и редактированием звука для самых разных целей, включая подкастинг и видеоклипы.

РЕДАКТИРОВАНИЕ ИЛЛЮСТРАЦИЙ И ИЗОБРАЖЕНИЙ

Gimp Блики на стекле

«К сожалению, наш сериал подходит к концу», – подумал **Майкл Хэммел** (Michael J Hammel) и сверкнул стеклянными глазами...

МЕСЯЦ НАЗАД

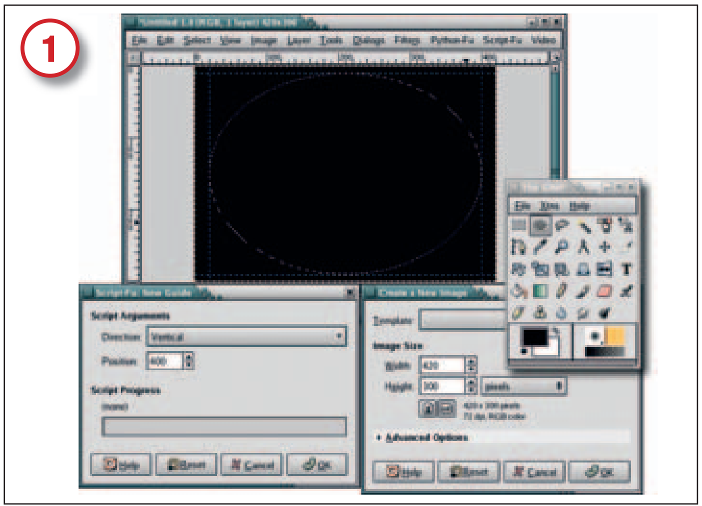
На предыдущем уроке я рассказывал, как собрать народ афишей оскароносного фильма – с направленным светом и тенью (благодаря Gimp).

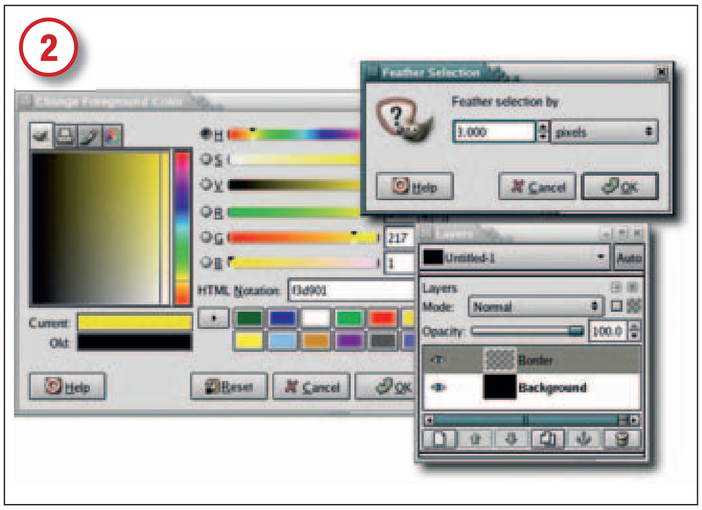
Световые эффекты очень полезны для имитации объёма в изображениях. Отражённый свет подразумевает некоторую вещественную поверхность, а тень позади этой поверхности завершает естественный порядок: свет, объект, фон. Можно добиться объёмного эффекта и без применения тени, просто придав отблеску необходимые очертания. В процессе урока мы разберём использование простых отражений на закруглённой стеклянной поверхности овального логотипа несуществующей фирмы.

Так как стекло гладкое, у наших фигурных отражений будут чёткие края. Но вообще-то отражение зависит от текстуры поверхности – например, на поцарапанном металле края отражений будут зазубренными и неровными. Мы возьмём одиночный источник света, дающий чёткое белое отражение. Он не обязательно должен быть белым, вы можете использовать обесцвеченные изображения (вообразите оконный проём или чьё-то лицо под белыми отсветами). С таким же успехом вы можете добавлять дополнительные источники света созданием более ярких узких отблесков слева или справа, или окрашенного отблеска сверху через всё стекло.

Техника проста в изучении, но требует некоторого терпения и практики. Суть состоит в нахождении правильной формы отражения и верного способа выемки выделенной области – выемка как раз и придаёт отражению форму. Обратите особое внимание на работу с линиями. Когда линия конвертируется в выделенную область, две ее крайние точки должны быть соединены, а для данного урока важно учитывать, где проходит прямая, соединяющая эти точки.

Прежде чем начать, я должен сообщить вам грустную весть: да, читатель, после трёх лет развлечений и игр (а также 10-летнего слежения за *Gimp*), это последняя статья в моей серии уроков по *Gimp* для *LXF*. Каждая минута работы была для меня в радость, а команда *LXF* щедро помогала донести до вас знания о *Gimp*, какие я только мог наскрести, все до последнего грамма. Я не бросаю LXF, но настало время поиска новых тем. Унывать не стоит – скоро выходит моя новая книга о *Gimp*! Следите за Starch Press, название книги *The Artists' Guide to Gimp Effects*. Увидимся!



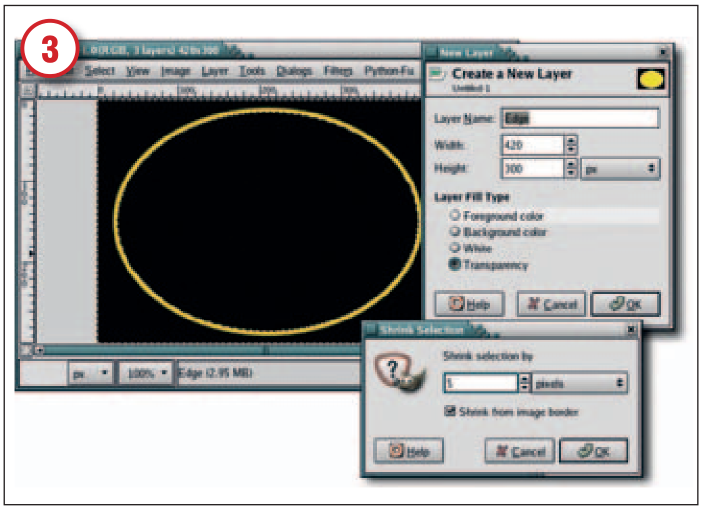


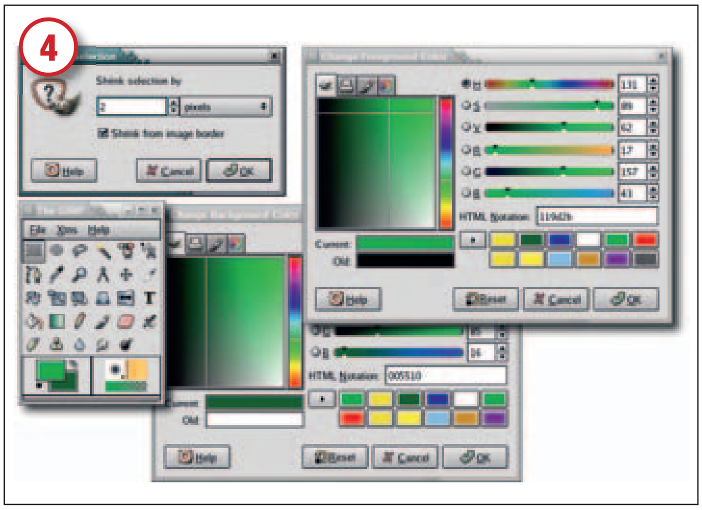
Готовим овал

Сначала убедитесь в использовании стандартных цветов и размеров: введите D (по умолчанию) в панели инструментов, цвета переднего плана и фона станут чёрным и белым соответственно. Начнём с чёрного холста (File > New – Файл > Создать) – цвет переднего плана в Дополнительных параметрах (Advanced Options). Данный урок легко масштабируется, для экспериментирования лучше всего подойдет стандартный холст размером 420x300 при разрешении 72 dpi. Если вам понадобится печатная версия проекта, весь процесс нетрудно будет повторить для большего холста с более высоким разрешением. Теперь добавим вертикальные направляющие (Image > Guides > New Guide – Изображение > Направляющие > Добавить направляющие) на 20 и 400 пикселей и горизонтальные – на 10 и 290 пикселей. Выберите инструмент эллиптического выделения в панели инструментов и протащите указатель мыши по диагонали от верхнего левого пересечения направляющих, чтобы выделяемый эллипс точно вписался в прямоугольник.

Добавим цвет

Растушуйте овальное выделение на 3 пикселя (Select > Feather – Выделение > Размыть). Добавьте прозрачный слой (Layer > New – Слой > Новый слой). Выделите название слоя в диалоге слоёв и контуров и измените его на «Граница». Откройте диалоговое окно выбора цвета переднего плана щелчком на соответствующем квадратике в панели инструментов. В поле HTML введите F3D901, нажмите OK, и передний план примет цвет яичного желтка. Перетащите цвет переднего плана из панели инструментов на выделение. Направляющие вам больше не понадобятся, поэтому избавьтесь от них в меню Image > Guides > Remove All Guides (Изображение > Направляющие > Удалить направляющие).



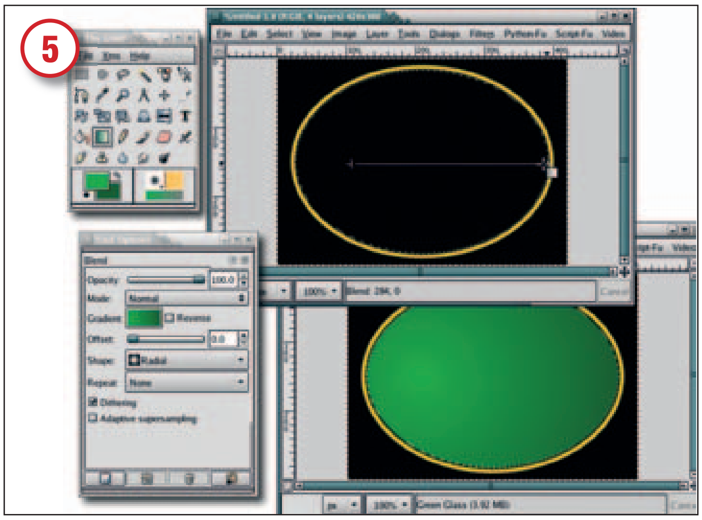


Завершим контур

Теперь у нас есть овал с жёлтыми краями. Добавьте к рисунку ещё один слой, назвав его «Край». Сожмите выделение на 5 пикселей (Select > Shrink – Выделение > Сжать). Измените цвет переднего плана на чёрный, введя D в окне изображения. Перетащите цвет переднего плана (то есть чёрный) на выделение.

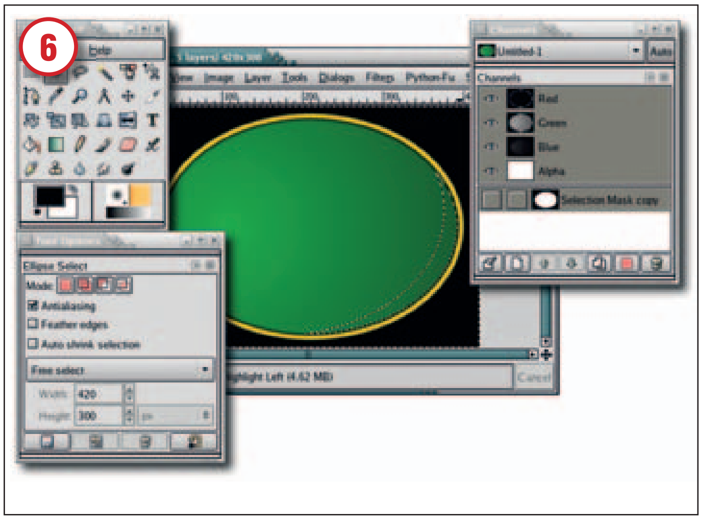
Круглый зелёный центр

Добавьте ещё один прозрачный слой (надеюсь, вы уже поняли, что слои – ключ к профессиональной работе в *Gimp*). Назовите слой «Зелёное стекло». Сузьте выделение на два пикселя и снова откройте диалоговое окно выбора цвета переднего плана. На этот раз введите в поле HTML 119D2B и нажмите OK. Затем, щёлкните на квадратике фонового цвета в панели инструментов, и в HTML-поле появившегося диалогового окна введите 005510. Нажмите OK. Эти цвета дадут нам замечательный диапазон оттенков зелёного для градиентного смешения на следующем шаге.



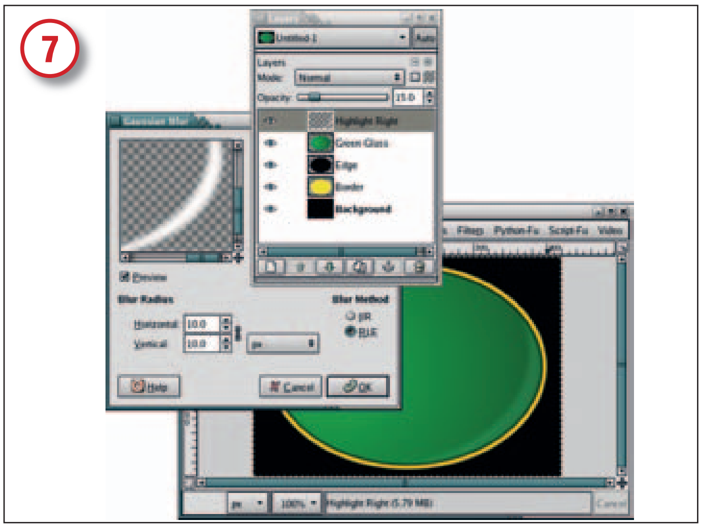
Формируем зелёный центр

Градиент придаст зелёному стёклышку ощущение твердости. Выберите в панели инструментов инструмент смешения. В диалоговом окне его настройки задайте градиент «от переднего плана до фона» (RGB) (убедитесь, что задано радиальное смешение). Непрозрачность должна быть 100%, а режим – нормальным. Это смешение понадобится вам только однажды. В окне изображения протащите курсор от левого фокуса овала (слева на средней линии) до его правого края, как показано на рисунке. В результате получится поверхность с переходом от нежно-зелёной до тёмно-зелёной окраски. Лёгкое изменение яркости уже создаёт впечатление округлой поверхности, а несколько отблесков усилят эффект.



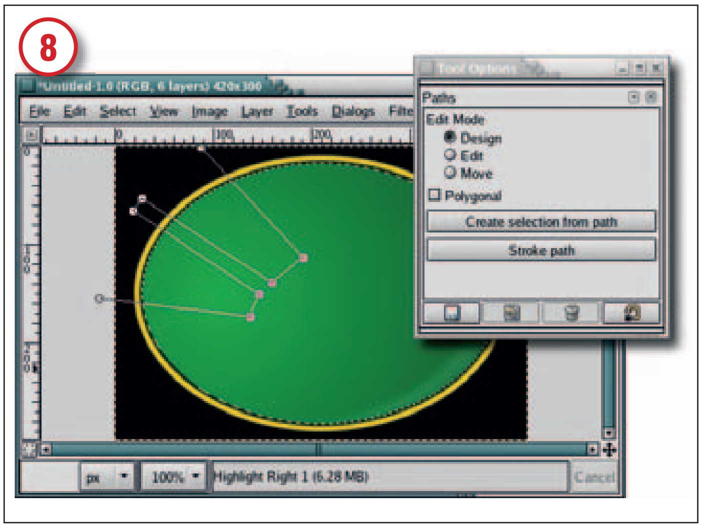
Начнём с первого отражения

Сохраните выделение в канале (Save > Save To Channel – Выделение > Сохранить в канале). Сохранение выделения в канале деактивирует все слои в диалоге слоёв и контуров, поэтому выделите слой «Зелёное стекло», чтобы вернуть ему активность. Добавьте новый прозрачный слой под названием «Подсветка справа» и переустановите цвета переднего плана и фона вводом D в окне изображения. Выберите в панели инструмент эллиптического выделения. В диалоговом окне его настройки нажмите кнопку Режима вычитания (третья кнопка слева в этом диалоговом окне, следующая за кнопкой режима). Протащите новый овал поверх существующего, чтобы вытеснить всё, кроме правого нижнего края, как показано на рисунке.



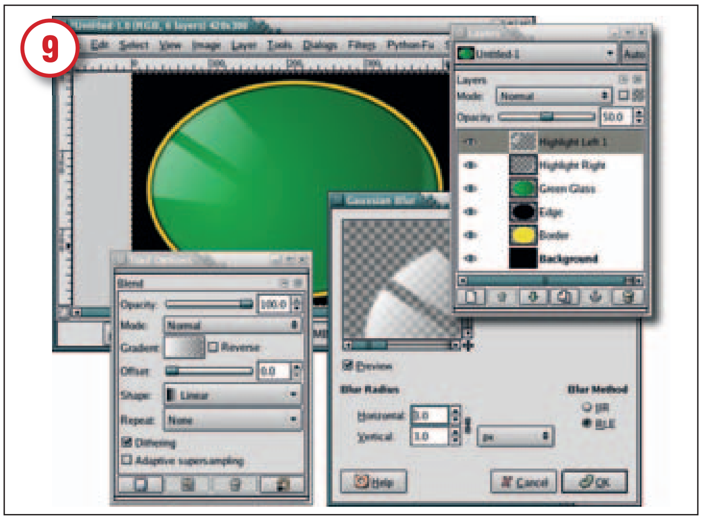
Размыть для большей реальности

Перетащите фоновый цвет (он должен быть белым) из панели инструментов на выделение. Отмените выделение (Ctrl+Shift+A), откройте диалоговое окно Гауссова размытия (Filters > Blur > Gaussian – Фильтры > Размыть > По Гауссу) и установите ширину размытия 10 пикселей и для горизонтального, и для вертикального направлений. Уменьшите непрозрачность слоя «Подсветка справа» до 15%. Если отблеск выглядит слишком резким, размытие всегда можно повторить, но лучше всё-таки сохранить некоторую резкость.



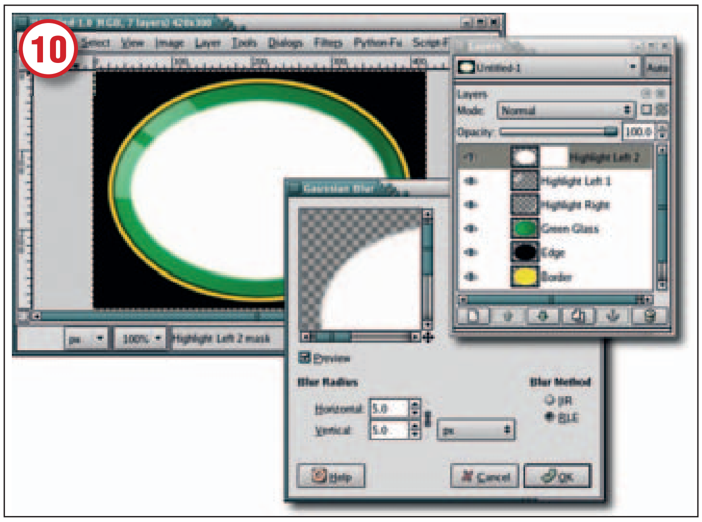
Ещё два отражения

Добавьте прозрачный слой и назовите его «Подсветка слева 1». Затем откройте диалоговое окно каналов (Dialogs > Channels – Диалоги > Каналы) и выделите сохранённый канал – последний в списке. Нажмите кнопку Channel To Selection (Канал в выделение), затем вернитесь в диалоговое окно слоёв и выделите слой «Подсветка слева 1». Овальное выделение послужит внешней границей двум новым отражениям. Чтобы создать их, начнём с выбора инструмента Path (Линия) в панели инструментов. Щёлкните за пределами выделения, затем прочертите линию вроде изображенной на рисунке. Убедитесь, что прямая, соединяющая начальную и конечную точки линии, находится за пределами существующего выделения. Затем, удерживая клавиши Shift и Ctrl, выберите Create Selection From Path (Создать выделение по контуру) в диалоговом окне настройки инструмента. В результате будет выделена область, заключённая между овалом и начерченной линией.



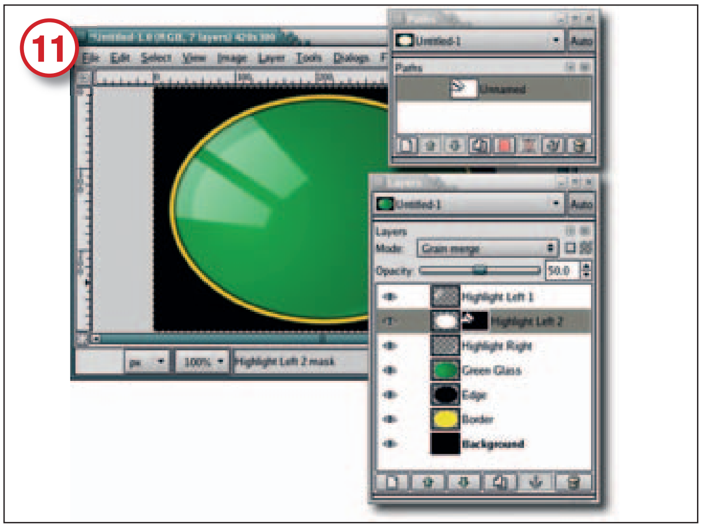
Задание размытия и непрозрачности

Введите в холсте X для обмена цветов переднего плана и фона – в результате фон станет белым. Выберите инструмент смешения из панели, и в диалоговом окне его настройки выберите градиент FG to Transparent (Передний план - прозрачный), а очертание установите в линейное. Протащите указатель от левого верхнего до правого нижнего угла выделенной области. Отмените выделение (Ctrl+Shift+A), ещё раз откройте фильтр Гауссова размытия и задайте трёхпиксельное размытие. Уменьшите непрозрачность слоя на 50%.



Высветите отражения

Теперь сделаем отражения светлее по направлению к центру: добавим новый прозрачный слой и назовём его «Подсветка слева 2». Снова вызовите сохранённое выделение (точно так же, как мы делали для слоя «Подсветка слева 1») и сожмите её на 20 пикселей (Selection > Shrink – Выделение > Сжать). Выделите новый слой, чтобы активировать его. В холсте введите D и X, чтобы сделать цвет переднего плана белым, затем перетащите цвет на выделенный участок для его заполнения. Снимите выделение (Ctrl+Shift+A), размойте слой на пять пикселей гауссовым фильтром и добавьте белую маску слоя (Layer > Mask > Add Layer Mask – Слой > Маска > Добавить маску).



Добавьте чёрного

Откройте диалоговое окно линий (Dialogs > Paths – Диалоги > Линии). Здесь должна присутствовать безымянная линия, с которой мы работали ранее линейным инструментом. Выделите линию в диалоговом окне, затем нажмите кнопку Path To Selection (Линию в выделение) внизу диалогового окна. Инвертируйте выделение (Select > Invert – Выделение > Инвертировать). Выделите маску слоя «Подсветка слева 2», заполните выделенную область чёрным и очистите. Размойте маску на пять пикселей гауссовым фильтром. Установите режим слоя на Убирать зернистость, а непрозрачность 50%. В диалоговом окне слоёв переместите слой «Подсветка слева 2» на одну позицию ниже слоя «Подсветка слева 1».

Введите текст

Теперь для полноты эффекта напишите имя компании. Введите D и X в окне изображения для установки белого цвета переднего плана. Выберите в панели текстовый инструмент, и в диалоговом окне его настройки выберите гарнитуру и размер шрифта для вашего названия. Щёлкните в окне изображения (откроется окно редактора), введите текст и закройте окно редактора. Выровняйте текст по центру инструментом перемещения. Отбросьте тень (Script Fu > Shadow > Drop Shadow – Скрипт Фу > Тень > Отбросить тень) кнаружи на 2 пикселя по горизонтали и вертикали с радиусом размытия 2 (убедитесь, что не нажата кнопка изменения размера). Нажав OK, добавьте тень. Поместите текстовый слой ниже обоих слоёв «Подсветки слева» в диалоге слоёв. Вот он – цветной овальный выпуклый стеклянный логотип. Согласитесь, приятный! LXF

ВЕКТОРНАЯ ГРАФИКА

Inkscape Работа с текстом

НА ДИСКЕ

ЧАСТЬ 5 Картина стоит тысячи слов – но и текст бывает необходим. **Дмитрий Кирсанов** покажет, как можно снабдить текстом ваши творения в *Inkscape*.

МЕСЯЦ НАЗАД

Мы рассматривали один из доступнейших способов освоения векторной графики: готовые фигуры Inkscape.

Лучшие векторные редакторы аккумулируют идеи и подходы из всех уголков мира компьютерной графики – растровых редакторов, CAD и инженерных пакетов, программ вёрстки, даже трехмерных и анимационных пакетов. Эти программы обладают общим свойством – все они способны производить графические изображения с текстом. Поэтому неудивительно, что хороший векторный редактор может служить в том числе и мощным инструментом для вёрстки и оформления текста.

Существуют два подхода к использованию текста в векторной графике. Один состоит в применении коротких текстовых строк – таких как логотипы, заголовки, слоганы. Такой текст называют рисованным; его размеры часто достаточно велики для того, чтобы были хорошо видны детали контура каждой буквы. Для работы с рисованным текстом нужна программа, способная отображать шрифт любой гарнитуры и начертания, дающая полный контроль над кернингом (расстояниями между символами) и позволяющая обрабатывать буквы как самостоятельные графические объекты, форму и стиль которых можно редактировать по отдельности.

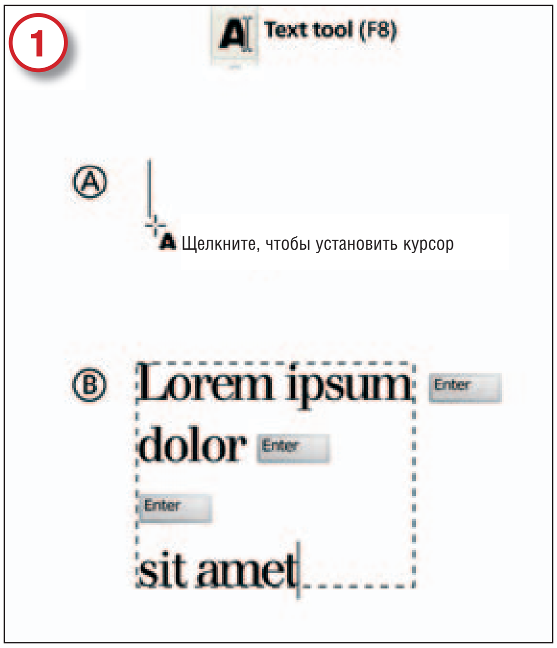
Текстовая заливка

С другой стороны, иногда в композиции нужно использовать текст довольно большого объёма, состоящий из многих строк и даже абзацев. В такой текстовой заливке контроль над каждой отдельной буквой требуется редко. Взамен от программы нужно другое: способность к автоматическому переносу строк, регулировка ширины колонки и расстояния между строками и, по возможности, способность правильно переносить слова со строки на строку. Особенно ценно, если ваша программа умеет связывать между собой отдельные текстовые поля так, что текст одного поля продолжается в следующем.

Поддержка текстовой заливки в векторных редакторах появилась сравнительно поздно. Стандарт SVG 1.1, созданный в конце 90-х, отражал положение дел на тот момент: все векторные редакторы прекрасно работали с рисованным текстом, но текстовая заливка была для многих недоступна. Соответственно, SVG 1.1 поддерживал однострочные текстовые объекты, текстовые фрагменты с разными стилями, произвольный кернинг и даже поворот отдельных букв в строке. Но многострочного текста и переноса строк в этом стандарте не было.

Ранние версии *Inkscape* поддерживали стандарт SVG 1.1, дополняя его эмуляцией многострочного текста (но без автоматического переноса). Однако, начиная с версии 0.42, программа частично поддерживает также стандарт SVG 1.2 (до сих пор не ставший официальным), в котором есть заливка текста в любую форму с автоматическим переносом строк. Поддержка текста в *Inkscape* все ещё сыровата, но вполне работоспособна, а о её практической применимости вы сможете судить сами, дочитав до конца этот урок.

Даже если вы не можете сделать с текстовым объектом все, что вам нужно, его всегда можно конвертировать его в контур, редактировать по узлам и придать каждой букве ту форму, которая вас устраивает. Тем не менее, лучшие векторные редакторы предлагают широкий спектр возможностей для работы с текстом без превращения его в контур – а значит, без потери его способности к текстовому редактированию.

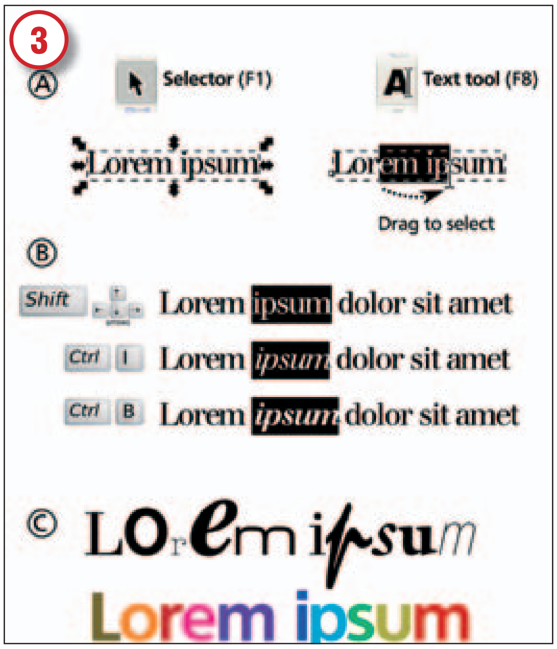


Создание текстового объекта

Чтобы создать простой текстовый объект (не заливку), выберите текстовый инструмент (F8), щёлкните в том месте холста, куда хотите поместить надпись, и начинайте печатать (A). Чтобы начать новую строку, нажмите Enter один раз, для вставки пустой строки – дважды (B). Для перемещения по тексту пользуйтесь обычными клавишами – стрелками, Home, End. Какое бы количество строк вы ни ввели, ваш текст остается единым текстовым объектом, который можно трансформировать целиком, выделив его соответствующим инструментом (Selector, «стрелка»).

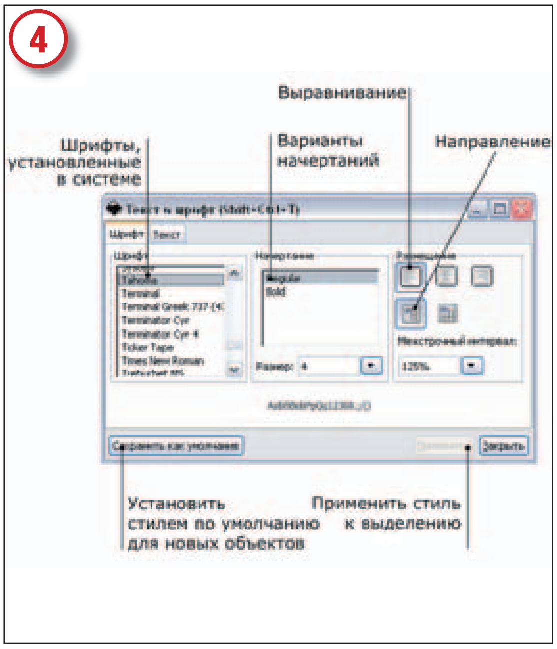
Смешение языков

Как все стандарты, основанные на XML, SVG использует Unicode. Одно из преимуществ такого подхода состоит в том, что в *Inkscape* можно смело смешивать любые языки, не беспокоясь о выборе верных кодировок и шрифтов. Если только в вашей Linux-системе установлен хоть один шрифт для какого-либо языка – *Inkscape* сможет отобразить текст на нем. Здесь приведён фрагмент файла **i18n.svg** из папки примеров *Inkscape*.



Выделение текста

Обычный нструмент выделения (Selector) может выделять только текстовый объект целиком. Напротив, инструмент Текст (Text) может выделять часть текста – текстовый фрагмент (A). Все горячие клавиши действуют при этом, как им и полагается: Shift+стрелки или протаскивание курсора выделяют часть текста, Ctrl+A выделяет весь текст, Ctrl+B и Ctrl+I делают выделенный фрагмент полужирным или курсивным (B). Любой другой стиль (например, цвет заливки) также применяется только к выделенному фрагменту. На приведённом рисунке для раскраски букв в разные цвета я воспользовался панелью Swatches (C).



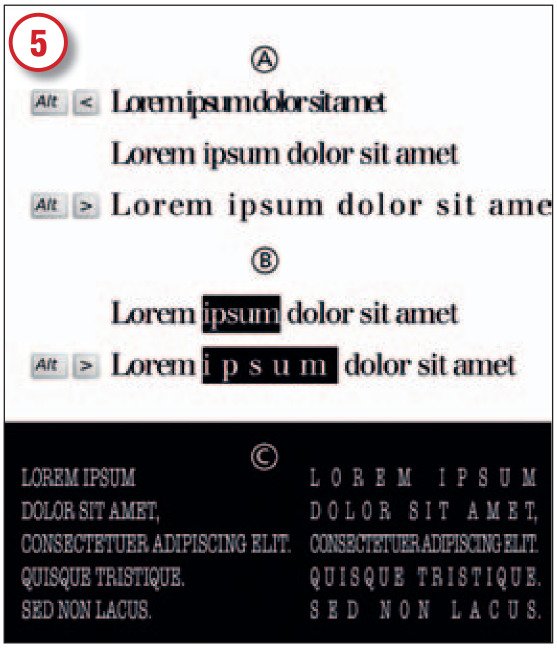
Диалоговое окно настройки текста и шрифта

Важнейший инструмент обработки текста – диалоговое окно «Текст и шрифт» (Ctrl+Shift+T). Здесь происходит выбор гарнитуры и начертания шрифта, размера, выравнивания и расстояния между строками. Вкладка «Текст» (Text) позволяет редактировать содержание текстового фрагмента (хотя в большинстве случаев удобнее делать это прямо на рисунке). Выбранный текстовый стиль можно сделать образцом для всех создаваемых текстовых объектов, нажав кнопку «По умолчанию» (Set As Default).

ПОДСКАЗКИ

- Одна из частей стандарта SVG, которую *Inkscape* пока не поддерживает — SVG-шрифты, хранящиеся прямо в SVG-документе. В настоящее время *Inkscape* может работать только с теми шрифтами, которые установлены в операционной системе. Если вы получили файл с встроенными SVG-шрифтами, то для его правильного отображения соответствующие шрифты придется доустановить. Для полной уверенности в том, что ваш *Inkscape*-документ будет правильно воспроизведён где угодно, проще всего конвертировать все текстовые объекты в контуры.
- Чтобы быстро удалить все ручные керны из текстового объекта (например, если вы решили поменять гарнитуру шрифта), воспользуйтесь командой «Текст > Удалить ручной кернинг» (Text > Remove Manual Kerns).
- Работа с текстовыми заливками в *Inkscape* основана на предварительной версии стандарта SVG 1.2, которая пока не поддерживается другими программами и вполне может измениться, прежде чем стандарт будет окончательно утвержден. Если вы намереваетесь просматривать свои SVG-файлы вне *Inkscape*, то лучше преобразовать все текстовые заливки или в контуры, или в обычный текст с жестким разбиением на строки («Текст > Конвертировать в текст» (Text > Convert To Text)). Оба способа сохраняют внешний вид текста, но при этом теряются некоторые из возможностей его редактирования.
- В диалоге выравнивания (Align and Distribute, Ctrl+Shift+A) есть четыре команды для выравнивания и распределения текстовых объектов. Эти команды выравнивают каждый текстовый объект по его базовой линии (той линии, на которой стоят буквы в строке) , а не по границам букв.

>>

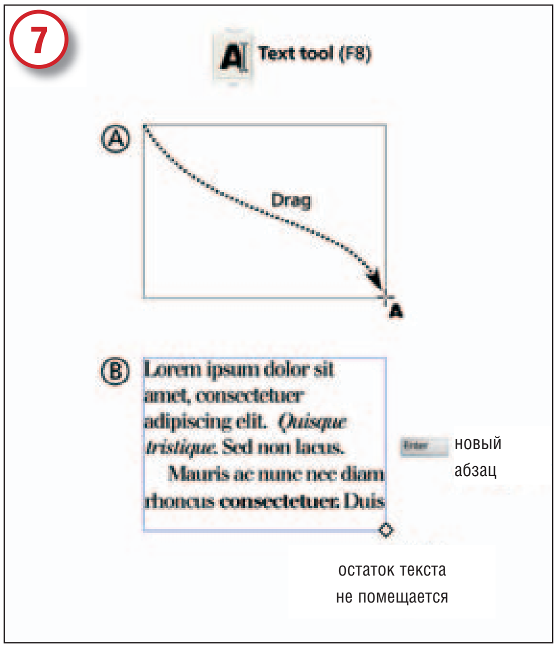


Межбуквенные интервалы

Диалог «Текст и шрифт» важен, но не полон: некоторые текстовые функции *Inkscape* доступны пока только через «горячие клавиши». Одна из них – интервалы между буквами. Комбинация клавиш Alt+< сближает буквы, Alt+> раздвигает их (A). Изменять интервалы можно как во всём текстовом объекте, так и в его выделенной части (B). Например, таким образом можно выровнять все строки текстового объекта по ширине (C).

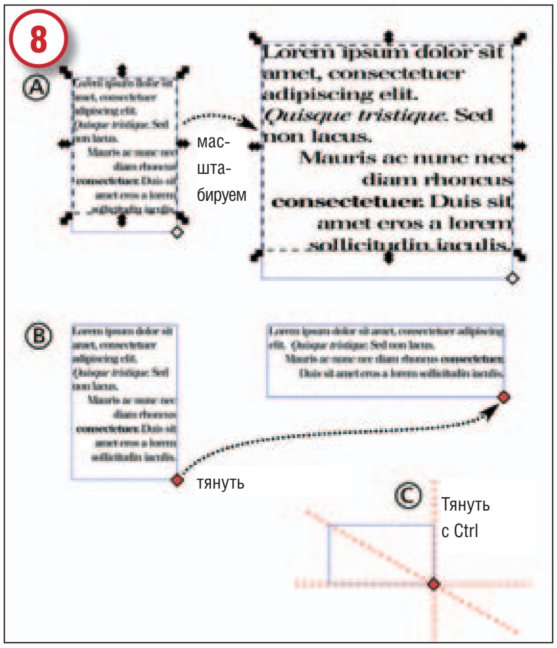
Кернинг

Другая текстовая настройка, доступная только с клавиатуры – кернинг. Поместите текстовый курсор перед какой-нибудь буквой и нажмите Alt+стрелку (A). Все символы после курсора до конца текста сдвинутся в направлении стрелки. Таким образом можно добиваться визуально равномерного размещения букв (B) или же создавать причудливые эффекты (C). Если нажимать Alt+стрелки при выделенном фрагменте, этот фрагмент будет смещаться относительно окружающего его текста (иначе говоря, равные по величине и противоположные по знаку керны будут вставляться перед и после выделенного фрагмента).



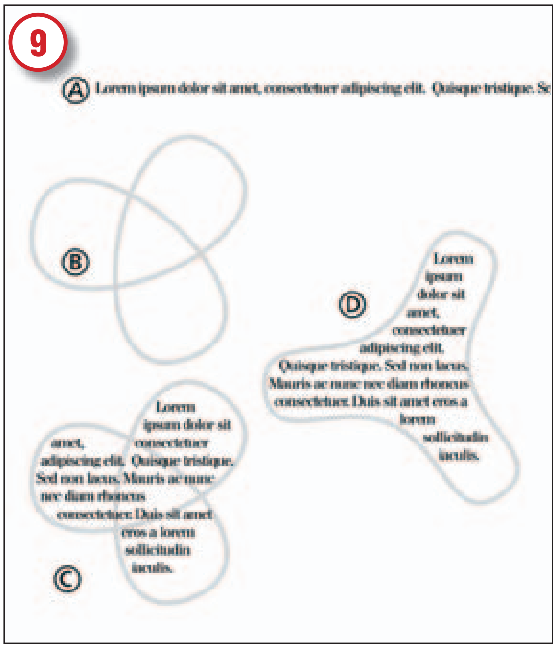
Текстовая заливка

До сих пор мы не упоминали о текстовой заливке с автоматическим переносом строк. Вот как она делается. Включите инструмент Текст и протащите курсор в любом месте рисунка (только не над другим текстом) (A). Возникнет прямоугольник с курсором в левом верхнем углу. Теперь можно начинать печатать; достигнув правого края прямоугольника, строки будут автоматически переноситься. Клавиша Enter начинает новый параграф. Текст, который выходит за нижний край прямоугольной рамки, не виден, но его можно вытащить на свет, увеличив размер рамки.



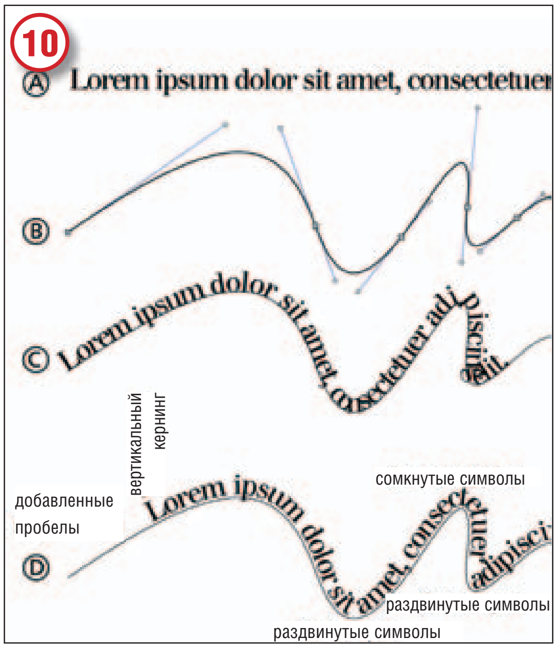
Трансформация заливки

Созданная таким образом текстовая заливка является единым объектом, включающим в себя как сам текст, так и рамку, в которую он залит. Поэтому инструмент выделения (Selector) трансформирует оба компонента одинаково (A). Чтобы растянуть или сжать только рамку, не меняя шрифт текста, перетаскивайте рукоятку в правом нижнем углу (B). При работе с этой рукояткой доступны все клавиатурные комбинации инструмента Прямоугольник (Rectangle). Например, протаскивая курсор при нажатой клавише Ctrl, можно перетягивать рамку только по одному измерению, зафиксировав другое (C).



Заливка в фигуры

Чтобы создать текстовую заливку непрямоугольной формы, нужно привязать текст к контуру. Выделите любой текст (A), затем контур или фигуру (B), и выберите «Текст > Залить в рамку» (Text > Flow Into Frame) (Alt+W). И текст, и фигура, в которую он помещён, останутся самостоятельными объектами, но будут связаны (C), так что перемещение, изменение размеров или формы рамки будет влиять также на текст внутри неё (D). Вытащить текст из рамки, превратив его в одну строку, можно командой «Текст > Извлечь» (Text > Unflow).



Изогнутые строки

Часто бывает нужно изогнуть текстовую строку вдоль кривой линии. В *Inkscape* это делается легко. Выделите текстовый объект (A) и контур (B), и отдайте команду «Текст > Пустить по контуру» (Text > Put On Path) (C). При этом и текст, и контур полностью сохраняют способность к редактированию. Попробуйте передвинуть текст вдоль контура, добавляя пробелы в начало; сдвинуть его перпендикулярно контуру, добавив вертикальный кернинг перед первой буквой; отрегулировать плотность букв на крутых поворотах с помощью горизонтального кернинга.

Штрих и заливка

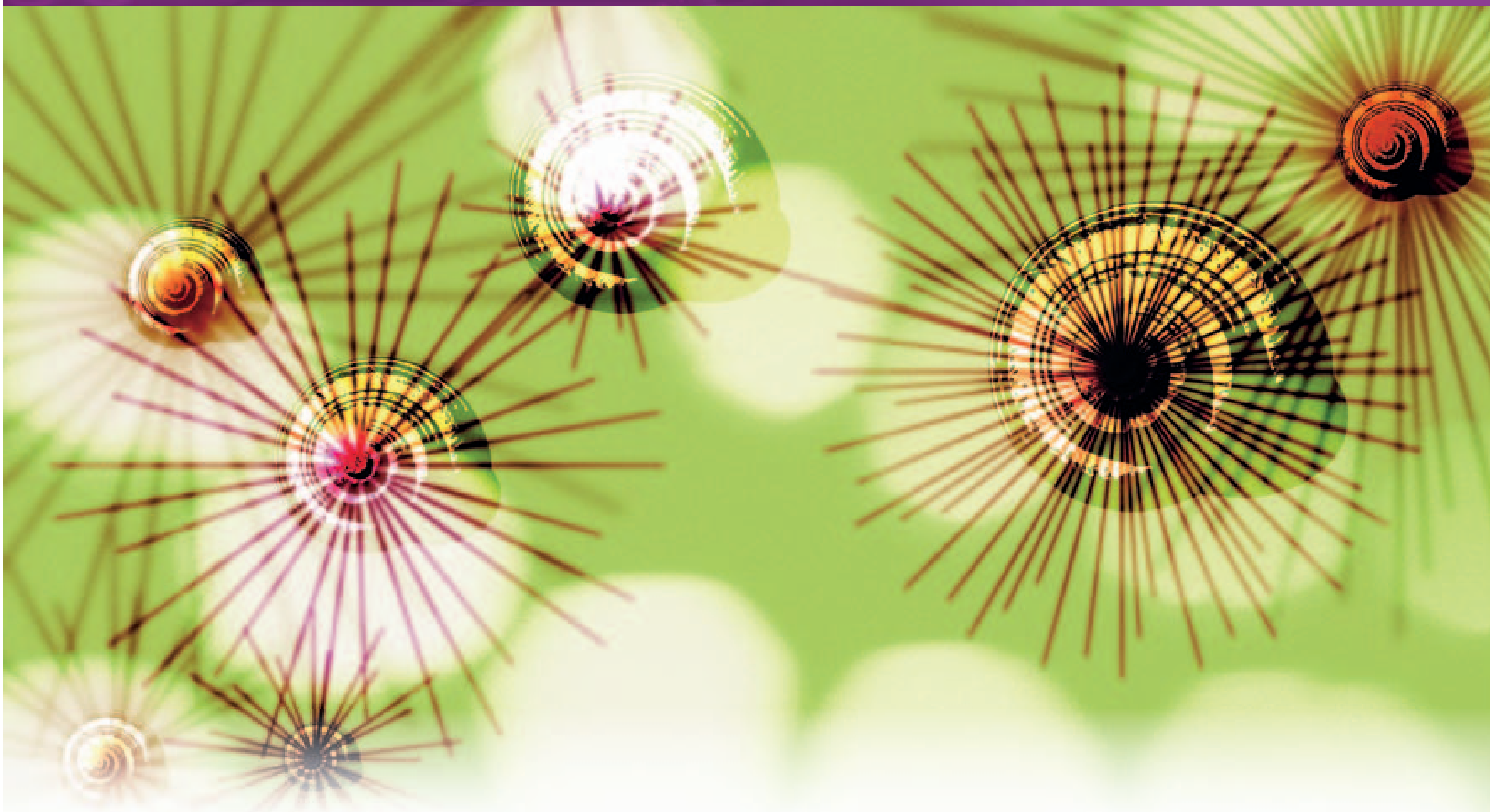
Как и любые другие объекты в *Inkscape*, текстовый объект может иметь штрих и/или заливку (A). Слишком широкий штрих, однако, может нарушить видимую форму букв, частично закрыв заливку (B). Решить задачу можно так: создайте копию текста, подложите её под имеющийся текст (Page Down), и примените штрих только к этой копии. В этом случае верхняя копия текста с заливкой, но без штриха, закроет внутреннюю половину штриха нижней копии, и текст будет выглядеть гораздо лучше (C).

Помимо слов

Если возможностей текстового инструмента *Inkscape* недостаточно, текстовый объект в любой момент можно превратить в контур и делать с ним всё, что вздумается. При таком превращении теряется возможность его текстового редактирования, поэтому лучше делать это только тогда, когда содержимое текста менять уже не придется. После этого вы можете, например, редактировать буквы-контуры по узлам (A), упрощать их (B) или применять к ним эффекты-расширения (C), которые мы будем изучать на следующем занятии. LXF

ЧЕРЕЗ МЕСЯЦ

Расширения *Inkscape* – это несложные скрипты, доступные из графического интерфейса, которые открывают новые возможности обработки объектов.



HARDCORE LINUX СЕРИЯ «ДЛЯ СПЕЦИАЛИСТОВ»

Мощные инструменты командной строки

Срубите GUI, вышвырните мышь из дома и дайте вашей графической супер-карте отдохнуть – «реальные пацаны» используют командную строку, утверждает **Пол Хадсон**.

МЕСЯЦ НАЗАД

Грэм Моррисон показывал, как создавать пакеты Autopackage.

Это четвёртый из наших уроков серии Hardcore Linux, и если вы досюда добрались, то вам почти пора наниматься в *Linux Format*. Заметьте, только почти, потому что ваши испытания на самом деле только начинаются. Вот как побываете под брюхом овцы, ослепите циклопа горящим бревном, перехитрите Цирцею и напишете собственный драйвер, тогда и будете готовы присоединиться к суперэлитной команде LXF.

А пока – сидите на коленях мастеров и учитесь, и в этом месяце вам откроются тайны командной строки. Причина, по которой храбрецы вроде нас овладевают командной строкой, та же самая, из-за которой Эдмунд Хиллари [новозеландец, впервые покоривший Эверест, в паре с проводником-шерпом Тенсингом Норгеем, – прим. перев.], по его словам, совершил восхождение на Эверест. Нет, не ради свежего горного воздуха, а просто потому, что Эверест есть.

Я хотел бы стать вашим Тенсингом Норгеем и помочь вам в восхождении к знаниям. Но, к сожалению, возможность слияния разумов исключается, ибо профанам Башни LXF недоступны, а Древнейшее Таинство Приобщения осложнено тем, что Mountain Dew в ближайшем универсаме на всех не хватит. Придется обойтись прозаическими бумагой и чернилами.

ШАГ 1 – СЕКРЕТЫ BASH

Стивен Борн (Stephen Bourne) написал первоначальный вариант командной оболочки Bourne для 7-й версии Unix в далёких семидесятых, и увидел Бог, что это хорошо. Но в 1987 году ее заменили улучшенной версией, названной Bourne-again Shell и большинству известной как *Bash*. Этот терминал уже много лет используется по умолчанию чуть ли не в каждом дистрибутиве Linux, так что новички даже не догадываются, что бывают и другие оболочки. Команда **ls -l /bin/sh** докажет вам, что самая основная оболочка на вашей системе – на самом деле просто ссылка на *Bash*, поскольку *Bash* – расширение sh, добавляющее новые функции, не нарушая работоспособность скриптов.

Bash еще функциональнее, чем можно вообразить. Всем нам доводилось пользоваться встроенными командами типа *cd* и *export*, а как насчёт *time*, *alias*, *for* и волшебного обратного апострофа? Даже управление заданиями, считавшееся абсолютно необходимым во времена больших компьютеров, – это вымирающее искусство. Ладно, я несколько отклонился от темы, дальше отклоняться не буду: настало время воскресить старые методы и поведать молодёжи, почему псевдонимы столь важны, сохраняя традиции наших Unix-предков. И первый шаг в этой революции – освоить *Bash*.

Меняем псевдонимы

Используемый по умолчанию набор команд в Unix довольно ограничен, но его можно обогатить с помощью различных параметров, передаваемых командам для модификации их работы. Кому нравится набирать команды, загибающиеся на три строки, поднимите руку! Конечно, это подвох – немногие люди, которые и вправду любят подобные вещи, вероятно, из-за артроза кистей рук поднять их не смогут.

Псевдонимы (aliases) – это простой способ создавать укороченные

команды в Linux. Например, можно определить псевдоним *la*, который запускал бы **ls -a**. На самом деле многие дистрибутивы поставляются именно с таким псевдонимом, установленным по умолчанию – если вы введёте *alias* в командной строке, то увидите список всех готовых псевдонимов. Чтобы создать собственный, введите

```
alias la='ls -a'
```

По идее, нужно вводить этот (и другие ваши псевдонимы) каждый раз после перезагрузки, но чтобы не увеличивать ваши ежедневные заботы, *Bash* позволит вам раз навсегда занести ваши псевдонимы в файл **~/.Bash_aliases**, да сам их и загрузит.

Условный ответ

Bash – мощная система разработки скриптов, способная проверять условия в операторах *if* и циклах не хуже других языков программирования. Скриптов я касаться не буду, но условные операторы могут очень пригодиться при использовании в псевдонимах.

Вот общий пример условного оператора:

```
if uname -a | grep «GNU/Linux» > /dev/null; then echo «Вы работаете в Linux»; fi
```

Работает он следующим образом: сперва исполняет **uname -a**, для получения информации о системе, а затем просеивает эту информацию через *grep*, пытаясь обнаружить строку «GNU/Linux». Заметьте, что вывод команды перенаправлен в **/dev/null**, чтобы результат *grep* не выводился на экран. Если *grep* найдёт строку, значит, наше условие истинно (true), и *Bash* выведет: «Вы работаете в Linux». Обратите внимание на использование точки с запятой: после условия и действия, но не после **fi** (if наоборот; по сути, то же, что и *endif*).

Мы можем усложнить оператор, добавив блок *else* – действие, которое будет выполнено, если условие вернёт значение «ложь» (false). Расширим предыдущий пример:

```
if uname -a | grep «GNU/Hurd» > /dev/null; then echo «Вы работаете в Hurd»; else echo «Вы работаете не в Hurd»; fi
```

Само условие может быть любой сложности. Попробуем, например, проверить, запущен ли *Firefox*, выполнив *ps aux* и перенаправив вывод на *grep* для поиска процесса *firefox*.

Однако, *grep* сам является программой, а значит, появится в выводе *ps*. И поскольку строка «*firefox*» содержится в «grep firefox», то окажется, что *Firefox* якобы запущен, хотя на самом деле мы всего лишь запустили поиск такой строки. Чтобы обойти эту проблему, применим *grep* дважды: сперва для проверки наличия *Firefox*, а затем, с ключом **-v**, для инвертирования условия, чтобы исключить из результата строку с *grep*. Итак:

```
if ps aux | grep firefox | grep -v grep > /dev/null; then echo «Firefox запущен»; fi
```

Совершенно неудобоваримо, так что заготовим псевдоним...

```
alias isff='if ps aux | grep firefox | grep -v grep > /dev/null; then echo «Firefox запущен»; fi'
```

Теперь, чтобы узнать, запущен ли ваш любимый браузер, достаточно выполнить команду *isff*.

Зацикливаем циклы

Условные операторы превосходны для основных задач, но истинная мощь *Bash* заключается в циклах – в них оператор неоднократно повторяется, обрабатывая новые данные. Иногда это просто выполнение команды несколько раз, например:

```
while (true); do sleep 1; date; done
```

Эта команда ежесекундно выводит на экран дату и время, пока не нажмут Ctrl+C. Но вам, скорее всего, захочется пройтись циклом по строкам файла или файлам в каталоге и что-нибудь с ними сделать.

Как, например, распаковать несколько файлов tar.bz2 одной командой? Новички вообразят, что можно скомандовать

```
tar xjf *.tar.bz2
```

Но при попытке это сделать возникает сообщение об ошибке: «tar: foo.tar.bz2: Not found in archive (не найден в архиве)». Дело в том, что *Bash* разворачивает регулярное выражение ***.tar.bz2** в список файлов каталога, фактически превращая введённую команду в такую:

```
tar xjf bar.tar.bz2 baz.tar.bz2 foo.tar.bz2
```

Когда вы передаёте утилите *tar* несколько файлов, первый из них рассматривается как имя архива, и *tar* попытается извлечь оттуда перечисленные далее файлы. Приведённая выше команда читается так: «извлечь baz.tar.bz2 и foo.tar.bz2 из bar.tar.bz2».

Значительно лучшее решение – использовать циклы: воспользуйтесь командой *ls*, чтобы получить список bz2-файлов в каталоге, и пройдитесь по ним в цикле, распаковывая каждый по отдельности. Итак:

```
for i in `ls *.tar.bz2`; do tar xjf «$i»; done
```

> «ИСТИННАЯ МОЩЬ BASH – В ЦИКЛАХ, МНОГОКРАТНЫХ ПОВТОРАХ ОПЕРАТОРА.»

Здесь используется оператор «`» (обратный апостроф, который обычно расположен на клавиатуре слева от цифры 1), он выполняет **ls *.tar.bz2** и вставляет её результат в команду на соответствующее место. Вы можете заставить *Bash* читать данные из файла, используя *cat*, например, так:

```
ls *.tar.bz2 > bzip_files
for i in `cat bzip_files`; do tar xjf «$i»; done
```

>>

ВКЛЮЧАЕМ ГЛАВНЫЙ ЭКРАН

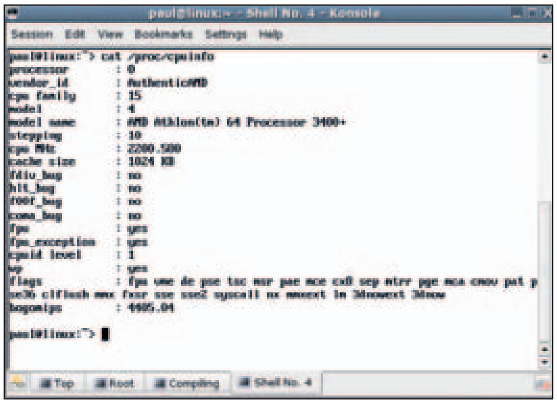
Работа в терминале воспринимается некоторыми программистами как возврат в восьмидесятые – точка взаимодействия с компьютером всего одна, и воспользоваться многозадачностью не так-то легко. Большинство систем Unix несколько сглаживают проблему, предоставляя возможность работать в нескольких виртуальных терминалах – нажатие <Ctrl+Alt+ с F1no F6> позволит войти в систему несколько раз и запустить разные задачи. Случай удалённого подключения – посложнее: если вы используете SSH на сервере и запустили долго выполняющуюся команду в интерактивном режиме, то как запускать следующую?

Если вы не придумали ничего лучше, чем «открыть ещё одно SSH-соединение», вас следует долго бить по голове толстенным руководством по Unix. Команда *screen* – это умножитель терминалов, то есть она преобразует один терминал в несколько и таким образом решает нашу проблему.

Сначала введите screen и нажмите <Enter>. Экран очистится, и вы увидите приглашение командной строки, ожидающее ввод; запустите команду top. Пока всё выглядит как обычно – вы можете подумать, что запуск screen совершенно ни к чему не привёл. Но нажмите **<Ctrl+a>+<c>** (нажмите и удерживайте <Ctrl>, нажмите <a>, отпустите <Ctrl> и нажмите <c>), и перед вами предстанет новый терминал. Команды управления в *screen* чувствительны к регистру: <Ctrl+A> не то же самое, что **<Ctrl+a>**. Старый терминал (и его команда top) всё ещё запущен, не приостановлен и не изменён, он работает в фоновом режиме, но мы сейчас находимся в виртуальном терминале 1. Чтобы вернуться назад, нажмите **<Ctrl+a>+<0>**, и вы обнаружите, что top всё ещё следит за работой вашего ПК. Для перехода обратно в новый терминал, нажмите **<Ctrl+a>+<1>**.

Есть множество других команд, которые вы можете освоить: **<Ctrl+a>+<n>** и **<Ctrl+a>+<p>** позволяют переходить на следующий и предыдущий терминал (переключаясь с последнего на первый и наоборот), **<Ctrl+a>+<A>** позволит задать имя терминалу (вместо общего для всех имени «Bash»), а **<Ctrl+a>+** выведет меню терминалов, по которому можно перемещаться с помощью стрелок курсора, не озадачиваясь точными номерами. Но важнее всех – **<Ctrl+a>+<d>**, которая отключает вас от сессии screen, но сохраняет её (включая терминалы и связанные с ними программы). Вы можете выйти из системы, прийти домой, набрать **screen -r**, чтобы подключиться к покинутой сессии, и вернетесь к тому же состоянию.

<< ШАГ 2 – МОГУЧИЕ КОМАНДЫ UNIX



***Konsole* позволяет держать открытыми сразу несколько терминалов. Переключаться между ними можно при помощи вкладок.**

Вторая часть нашего урока посвящена тому, как выжать всё из команд, которыми вы пользуетесь ежедневно – или, по крайней мере, должны бы пользоваться. Все команды Unix очень хорошо выполняют какую-то одну простую задачу, но чтобы предоставить опытным пользователям (вроде вас – по крайней мере, такими вы должны стать после наших уроков) максимальную гибкость, эти команды имеют множество параметров, чтобы изменять их поведение.

«Меньше» значит «больше»

Рассмотрим команду *less*. Вероятно, вы используете её так:

```
cat /etc/somefile | less
```

Она выводит на экран файл и позволяет вам перемещаться по нему с помощью стрелок курсора. Пользователи поумнее (это мы) вместо этого употребляют команду

```
less /etc/somefile
```

Эффект будет тот же. А если я скажу вам, что у less множество параметров, изменяющих ее работу? На них прямо-таки нехватает букв алфавита, а некоторые даже чувствительны к регистру символов! И это только параметры, которые вы передаёте в программу – а ведь после запуска *less* ею можно управлять с помощью различных комбинаций клавиш.

> «У LESS ТАК МНОГО ПАРАМЕТРОВ, ЧТО АЛФАВИТА НЕ ХВАТАЕТ.»

Сильно ли вас бесит сообщение об ошибке, говорящее, что в файле somefile.txt есть проблема в строке 780? Не все так плохо: вспомните, что команда *less* с параметром **-N** (заметьте: не -n) распечатает номера строк, и вы узнаете, в чем дело. А чтобы получить больше информации, можете попробовать ключ **-M** – в строке статуса будут выведены имя файла, диапазон строк и текущая позиция в процентах.

Можно передать *less* несколько имён файлов, и они будут открыты все за одну сессию. Например:

```
less /etc/passwd /mnt/backup/etc/passwd
```

ГРАФИЧЕСКИЕ ТЕРМИНАЛЫ

«Графические терминалы» – не оксюморон. Оказывается, использование приложений вроде *Konsole* может сэкономить ваше время, сделав работу за вас. Моя любимая функция – «monitor», это когда *Konsole* следит за активностью терминала и высвечивает сообщение о перемене состояния. Так что если вы ожидаете завершения длительного процесса компиляции, просто поручите *Konsole* проследить и доложить.

Когда *less* запустится, в её буфер будет загружен **/etc/passwd**, и в строке статуса появится сообщение «Файл 1 из 2». Для перемещения к следующему файлу (**/mnt/backup/etc/passwd**) используйте **:n**, а чтобы вернуться назад к первому – **:p**. Открыть следующий файл, сохраняя уже открытые два, можно взять **:e** и ввести имя файла, используя Tab для автодополнения, если потребуется. Чтобы закрыть текущий файл и убрать его из буфера less, пригодится **:d**.

Наконец, мощная клавиша «!». Если вы наберёте **!** с последующей командой, которую нужно выполнить, *less* запустит эту команду, покажет вам её вывод и вернётся к просмотру файлов. Бесполезное вроде занятие; но вспомните, что текущий файл обозначается «%». Допустим, вы просматриваете в *less* файл **/home/paul/somefile.txt** и сообразили, что хотели включить в него список файлов из каталоге **/etc**. Сделать это можно прямо из *less*:

```
!ls /etc >> %
```

Перенаправление **>>** припишет вывод команды *ls* к концу файла. Когда вы вернётесь в *less*, её буфер будет обновлён, чтобы показать результат программы, и вы можете продолжить чтение.

top-парад

Команда *top* – одна из тех, которые новички осваивают в первую очередь, поскольку это простейший способ узнать, какая программа сжирает все процессорное время в вашей системе. Но поначалу *top* может напугать: уйма данных вверху, далее 12 или больше столбцов информации, напичканной акронимами...

Столбцы, которые нам интересны – PID (идентификатор запущенного процесса), User (пользователь, запустивший процесс), NI (значение niceness, показывающее, с какой лёгкостью процесс отдаёт свои ресурсы) и сама команда. Остальные значения можно либо игнорировать, либо просто удалить из списка отображаемых полей. Для этого нажмите клавишу F, и вы увидите список всех отображаемых полей, вместе с клавишей, которую следует нажать для отключения поля. Например, «* N: %MEM = Memory usage (RES)» означает, что %MEM сейчас отображается (помечено звёздочкой), а нажатие n исключит это поле из списка. Выбрав нужные поля, нажмите Enter – и вернетесь к основному экрану.

Если вы нажмёте F (заметьте, что буква на сей раз заглавная), вы увидите примерно такой же список, но теперь сможете выбрать, какое поле будет использоваться для сортировки. Для пробы нажмите F (чтобы попасть на экран сортировки), a (выбор сортировки по PID), Enter (возврат в основной экран) и затем R (сортировка в обратном порядке – первыми отображаются PID с наименьшим значением).

top умеет не только показывать цифры: это мощнейший инструмент администрирования. Например, нажмите k и введите PID процесса, который нужно завершить (по умолчанию используется сигнал 15, который вежливо просит программу прекратить работу; в случае неудачи сигнал 9, так сказать, «вырубить питание»). Можно также нажать **<r>**, указать PID процесса и изменить его значение nice, и вашей системе гарантирован более высокий приоритет. Здесь есть одна хитрость – чем ниже значение nice, тем выше приоритет, самое низшее значение – 20, наивысшее – минус 19. Будьте внимательны при установке значений nice: повышать приоритеты разрешено только от имени суперпользователя, и любое значение ниже нуля требует особой осторожности.

Грызем grep

Держу пари, что я смогу по пальцам сосчитать людей, которые регулярно используют *grep* в ситуациях, отличных от этих двух:

```
somecommand | grep «foo»
```

или

```
grep «foo» *
```

Увы, при своей-то чудовищной мощности, *grep* обычно прикован к дебильным задачам, едва отвлекающим процессор от спячки. *Grep* тоже имеет столько параметров, что и не упомнишь, но некоторые всё же следует выучить:

- **-c** Подсчитывает число вхождений искомой строки.
- **-i** Поиск, не зависящий от регистра символов.
- **-l** Выводит на экран имена совпавших файлов (это строчная L).
- **-n** Отображает номер соответствующей запросу строки.
- **-r** Выполняет рекурсивный поиск по каталогам.
- **-v** Инвертирует условие поиска (строки, не содержащие цель поиска).

Их можно объединять, перенаправив *grep* на самого себя. Например, если нужно подсчитать число файлов, которые содержат слово «paul» (или «PAUL», «PaUl» и т.д.), кроме тех, которые имеют расширение .txt, вам следует сделать что-то подобное:

```
grep -ilr paul * | grep -cv «\.txt$»
```

Здесь мы используем пять из шести важнейших параметров, превращая сложную задачу в простой конечный результат: число соответствующих условию файлов. В этом примере **-ilr** означает, что выполняется поиск по файлам слов Paul, PAUL, PaUl и т.д., поиск проходит рекурсивно по вашей файловой системе (начните с корневого каталога, чтобы выполнить поиск по всему диску) и возвращает имена подходящих файлов. Всё это поступает на вход другого экземпляра *grep*, который использует флаги **-cv**, включающие режим подсчёта и поиск файлов, которые не соответствуют регулярному выражению «\.txt$». Обратный слеш здесь экранирует точку: без него мы получили бы специальный символ. Знак доллара в конце означает, что .txt должно находиться в конце имени файла, то есть foo.txt.bar не будет соответствовать выражению.

Если вам предстоит прошлепать по огромному тексту, настоятельно рекомендую использовать дополнительный ключ **--color**, чтобы grep подсвечивал в выводе соответствующие выражения.

Ищите и обрящете

Команда *find* печально знаменита своим коварством: никто не может правильно её использовать, не прочитав предварительно руководство. Впервые принимаясь за *find*, многие думают, что нужно попробовать нечто вроде **find <иголка> <стог_сена>**. Например, **find myfile.txt /etc** должна бы просматривать каталог **/etc** и его подкаталоги в поисках файла **myfile.txt**. Ха-ха! Когда я вижу новичков, набирающих подобное, я обычно хлопаю их по спине, говорю, что всё у них получится, и расхаживаю, невинно посвистывая.

Вместо того чтобы работать в соответствии со здравым смыслом, *find* работает как фильтр. Наиболее общая команда выглядит так:

```
find .
```

Первый параметр (путь, по которому выполняется поиск) – это «.», так что будет возвращён список всех файлов в текущем каталоге. Теперь сузим вывод при помощи параметров. Например, **-name** указывает, что *find* должна возвратить только файлы с указанным именем, скажем, myfile.txt или «*bar*», соответствующий файлам с «bar» где-нибудь в имени. Например:

```
find . -name «*bar*»
```

Мы могли бы добавить параметр **-size**, который ограничивает результаты поиска файлами, размер которых больше, меньше или равен заданному. Например:

```
find . -name «*bar*» -size +1M
find . -name «*bar*» -size -10k
```

Первая команда найдёт все файлы с нужными именами, которые больше 1 МБ, а вторая – те, которые меньше 10 КБ. Файлы, размер которых точно равен заданному (например, 1 МБ), не будут возвращены, если поиск выполняется со строгим условием «больше» или «меньше».

Параметр **-user** возвращает только файлы, принадлежащие указанному пользователю. Например, **-user paul** вернёт файлы, принадлежащие пользователю paul. В частности, это полезно с параметром **-not**, который может предшествовать любому параметру и выполняет инвертирование значения. Вот несколько примеров:

```
# все файлы, не принадлежащие пользователю paul
find . -name «*bar*» -not -user paul
# все файлы, кроме myfile.txt и тех, которыми владеет paul
find . -not -name «myfile.txt» -not -user paul
# аналогично первому примеру, но с двойным отрицанием для прикола!
find . -not -not -name «*bar*» -not -user paul
```

Ещё один небезынтересный фильтр – **-newer**, он возвращает все файлы, которые новее, чем указанный файл. Это очень удобно для скриптов резервного копирования: когда делаете копию, просто укажите произвольный файл, и он будет служить временной меткой, которую можно использовать в команде **find -newer**. Вы получите список всех файлов, которые изменились с момента создания этого файла. Вот нужная вам команда:

```
find . -newer /path/to/myfile
```

Последние штрихи

Есть целая группа параметров *find*, которые мы не затронули: действия. Каждый раз, когда find наталкивается на файл, соответствующий вашим критериям, она может что-то с этим файлом проделать. Например, **-ls** выведет на экран информацию об этом файле в стиле **ls -l**, а параметром **-exec** заставит *find* запустить для этого файла определённую команду. Скажем, вы хотите найти все текстовые файлы в своём домашнем каталоге и вывести их на экран. Это можно сделать так:

```
find /home/someuser -name «*.txt» -exec cat {} \;
```

> «НЕСМОТРЯ НА ЧУДОВИЩНУЮ МОЩЬ, GREP ОБЫЧНО ЗАНЯТ ЗАДАЧАМИ ДЛЯ ДЕБИЛОВ.»

Фигурные скобки **{}** заменятся именем соответствующего файла, и при каждом совпадении будет выполняться *cat*. Конструкция **\;** в конце необходима, потому что *find* считает всё, что после **-exec**, выполняемой программой, пока не наткнется на точку с запятой. Но если вы просто наберете **-exec cat {};**, то *Bash* в процессе обработки строки эту точку с запятой удалит, и *find* пожалуется на ошибку.

Альтернативой **-exec** является **-ok**, который работает точно таким же образом, но запрашивает ввод пользователя перед выполнением каждой команды. Нажмите **<y>** для выполнения команды (опять-таки, для каждого соответствующего файла), или **<n>**, чтобы пропустить её.

К сожалению, на этот месяц больше места нам не досталось, и я сумел лишь обсудить *Bash* и четыре самых популярных команды Unix. Мы надеемся, что наступит день, когда вы полюбите командную строку с той же страстью, что и мы – или, по крайней мере, не так будете пугаться пролетающего во время загрузки текста. LXF

ПОЛЕЗНЫЕ СОВЕТЫ

- Меняйте права доступа мгновенно: используйте **chmod --reference /path/to/somefile.txt ***, и у всех файлов в текущем каталоге будут такие же права, как у **somefile.txt**.
- Команда **tail -f /var/log/somefile.log** загрузит последние десять строк лог-файла и будет обновлять вывод по мере появления новых.
- Вместо того, чтобы повторно набирать команду в ожидании события, используйте команду *watch*. Например, **watch ls -l** будет запускать **ls -l** каждый две секунды и выводить полученный результат на экран.
- Если вам нравится мониторинг активности в *Konsole* и хочется получить то же самое в screen, просто используйте

```
<Ctrl+a>+<M>.
```

- Хороший снимок рабочего стола *X* получится с помощью следующей жуткой команды:

```
chvt 7; sleep 3; import -display :0.0 -window root screenshot.png
```

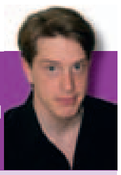
ЧЕРЕЗ МЕСЯЦ

Bash вернется для отмщения, а также для управления заданиями. Но в основном для управления заданиями.

MYSQL 5.0 SPECIAL ПРОГРАММИРОВАНИЕ СЦЕНАРИЕВ

PHP Представления

ЧАСТЬ 3 Живите легко! Сократите объемы SQL-кода и повысьте производительность. **Пол Хадсон** представляет себе последствия...



МЕСЯЦ НАЗАД

Благодаря хранимым процедурам, мы подобрались к СУБД так близко, насколько это возможно (если не вскрывать корпус компьютера).

Хотя структура вашей БД, скорее всего, меняется не каждые пять минут, существует множество способов подать одни и те же данные. В сложной системе доступ ко всей базе данных (и записи в нее) может быть предоставлен только администратору. Простым смертным приходится довольствоваться лишь чтением какой-либо таблицы или даже ее части.

Если вы не можете найти реальный пример, подумайте об отделе кадров. Его сотрудники должны легко находить ваше имя, должность и так далее. Но ваш домашний телефон или реквизиты банковского счета – это не их дело. Такая информация является личной, и доступ к ней предоставляется только старшему персоналу. Можно также представить себе еще один уровень доступа, которым обладают все сотрудники компании, позволяющий, например, узнать имя, должность и внутренний номер телефона.

Так что, даже если в вашей таблице всего одна строка данных, вы имеете три разных способа их подачи: полный доступ, ограниченный доступ и малый доступ. *MySQL 5.0* предоставляет средства для программирования этих разграничений в самой базе данных, так что каждый пользователь будет иметь доступ лишь к одному представлению.

Представление начинается

Думайте о представлении как о фиксированном SQL-запросе, который определяет, что может увидеть пользователь в таблице. В этом учебнике мы будем работать с таблицей employees (сотрудники):

```
CREATE TABLE employees (ID INT NOT NULL AUTO_
    INCREMENT PRIMARY KEY, FirstName VARCHAR(100),
    LastName VARCHAR(100), Age INT, SSN VARCHAR(30),
    IBAN VARCHAR(100));
```

В базе данных хранится имя и возраст человека (вполне безобидная информация), номер социального страхования (SSN) и International Bank Account Number (IBAN) – определенно приватные данные. Мы хотим, чтобы другие сотрудники видели только имя и возраст. Отделу кадров понадобится и SSN, а бухгалтерии – еще и IBAN. Чтобы сделать это, нам потребуются три представления (view): 'restricted' (ограниченный), 'HR' (отдел кадров) и 'accounts' (бухгалтерия).

Представления создаются при помощи оператора **CREATE VIEW**, за которым следует стандартный запрос **SELECT**:

```
CREATE VIEW restricted AS SELECT FirstName, LastName,
    Age FROM employees;
```

Представления выглядят как виртуальные таблицы: они не занимаются реальным копированием данных из employees и не появляются в выводе команды **SHOW TABLES**. Чтобы увидеть представления, необходимо выполнить **SHOW FULL TABLES**. Ее вывод будет разделен на две колонки. Строка **BASE TABLE** обозначает стандартную таблицу *MySQL*, **VIEW** соответствует представлению.

Однако в смысле выполнения запросов представления ведут себя как самые обычные таблицы. Например, чтобы прочитать все строки таблицы employees от имени «ограниченного» пользователя, выполните: **SELECT * FROM restricted;**. Вы также можете добавить некоторые условия, например:

```
SELECT * FROM restricted WHERE Age > 40;
SELECT * FROM restricted LIMIT 3;
SELECT MAX(Age) FROM restricted;
```

Последняя строка представляет особенный интерес, поскольку в ней мы «фильтруем фильтр»: представление restricted является подмножеством employees, а мы выделяем из него одну-единственную колонку. По сути, это эквивалентно:

```
SELECT MAX(Age) FROM (SELECT * FROM employees);
```

Но что мешает пользователю прочитать данные из employees, а не из restricted? Пока – ничего. Пока...

Ограниченное представление

MySQL весьма гибок в отношении пользовательских прав на доступ к таблицам. Эта же гибкость сохраняется и для представлений, а значит, мы можем позволить людям читать только одно представление, или же читать одно, а писать – другое. Прежде всего, нам потребуется пользовательская учетная запись, так что подключитесь к *MySQL* как root и наберите:

```
USE MySQL
INSERT INTO user (Host, User, Password, ssl_cipher, x509_
    issuer, x509_subject) VALUES ('localhost', 'peon',
    PASSWORD('p30n'), '', '', '');
```

В зависимости от версии *MySQL* и режима работы (strict или не-strict), вам может потребоваться поиграть с **ssl_cipher**. В любом случае, теперь у нас есть новый пользователь, которому нужно предоставить право на чтение представления:

```
GRANT SELECT ON your_db_name.restricted TO
    peon@localhost;
FLUSH PRIVILEGES;
```

В итоге пользователь peon получит доступ только к таблице или представлению 'restricted' в БД **your_db_name**. Теперь отсоединитесь от *MySQL* и вновь зайдите как новый пользователь. Попробуйте выполнить следующее:

```
SELECT * FROM employees;
```

Вы получите сообщение об ошибке: «SELECT command denied to user 'peon'@'localhost' for table 'employees'». На самом деле, выполнив **SHOW TABLES**, вы увидите только представление 'restricted' – *MySQL* делает вид, что никакой таблицы employees не существует. Чтобы предоставить пользователю peon чуть больше прав, опять зайдите как root и выполните:

```
GRANT SELECT,INSERT,DELETE,UPDATE ON test.restricted TO
    peon@localhost;
FLUSH PRIVILEGES;
```

Это позволит пользователю манипулировать данными таблицы, не боясь навредить ей самой.

Производные

Производные колонки (derived columns) позволяют хранить в вашем представлении синтетические данные. Обычная колонка просто читает значение поля таблицы и помещает его в представление, тогда как производная колонка читает некоторое поле, выполняет преобразование и лишь затем помещает данные в представление. Умно!

Один из моих любимых принципов БД – атомарность. Я предпочитаю хранить все поля по отдельности, независимо друг от друга, и вычислять нужные величины «на лету» по мере надобности. Например, если меня интересует число страниц, которые пользователь посетил в местной интранет-сети, я сохраню запись о каждой странице, а также сопутствующую информацию (дату, время, ...) в отдельности, а затем запущу запрос, который посчитает число строк – это и будет количество страниц, посещенных данным пользователем.

Вашему боссу это может не понравиться – ведь он-то хочет видеть в базе данных одно число, а не отдельные запросы! С представлениями и производными колонками мы можем легко синтезировать эти данные и предоставить боссу на блюдечке.

Вычисляемые колонки

В качестве иллюстрации, рассмотрим следующую таблицу для несуществующего файла журнала:

```
CREATE TABLE page_requests (ID INT NOT NULL AUTO_
    INCREMENT PRIMARY KEY, UserID INT, PageID INT,
    RequestTime INT);
```

После занесения в нее кое-каких данных, можно создать «боссово представление» следующим образом:

```
CREATE VIEW total_requests AS SELECT UserID, Count(*) AS
    RequestCount FROM page_requests GROUP BY UserID;
```

Здесь мы создаем производную колонку, основываясь на числе запросов, которые сделал пользователь. Вы не можете обновить ее или удалить какую-либо строку на основе ее данных, поскольку на самом деле это не колонка, а вычисление, которое *MySQL* производит по ходу работы.

Это – самый распространенный способ использования производных колонок. Например, отдел кадров может не интересовать, когда именно человек был в отпуске, но его сотрудникам важно знать, сколько у него осталось неиспользованных дней. Производные колонки также можно использовать для замещения значений колонки нужными нам величинами. Например, мы использовали представления, чтобы ограничить доступ к SSN и IBAN. Возможно, лучшим решением было бы поместить в этих колонках служебное сообщение, указывающее, к кому следует обратиться за теми или иными сведениями. Вот соответствующий код:

```
CREATE VIEW restricted_contact AS SELECT FirstName,
    LastName, Age, «Обратитесь к Иванову из отдела кадров» AS SSN,
«Обратитесь к Петрову из бухгалтерии» AS IBAN FROM employees;
```

Вид сверху

Представления – прекрасное средство для обеспечения безопасности, но они могут также оказаться полезными при работе со сложными SQL-запросами, оперирующими с несколькими источниками данных. Вся «соль» в том, что пользователь может работать с представлением как с таблицей, свободно выбирая колонки и способ упорядочения данных. Длинные SQL-запросы долго пересылаются, обрабатываются и служат прекрасной почвой для разведения различных ошибок. Представления позволяют инкапсулировать эти сложные запросы в простые конструкции SELECT, которыми пользователь может распоряжаться по своему разумению.

Допустим, у нас есть две таблицы: одна – для сотрудников и вторая – для записей о просмотренных ими страницах. Мы можем объединить их при помощи LEFT JOIN:

```
SELECT employees.*, count(page_requests.ID) AS
    PageRequests FROM employees LEFT JOIN page_requests
    ON employees.ID = page_requests.UserID
GROUP BY page_requests.UserID;
```

Это даст нам всю информацию о человеке, включая число посещенных страниц. Правда, запрос получился не такой уж короткий, и многие наверняка споткнутся, дойдя до **LEFT JOIN**. Используя представления, вы можете сократить его до **SELECT * FROM employees_complete** и забыть об этих проблемах.

Все, что мы видели до сих пор, позволяет сократить время разработки и снизить число ошибок в коде, однако, представления обладают еще двумя преимуществами: во-первых, они позволяют создать одну большую «супертаблицу» и подпредставления (subview), которые выбирают данные из обширного источника. Во-вторых, представления – это, по сути, функции, которые можно изменить в любое время.

Как я уже неоднократно говорил, в большинстве случаев представление действует как таблица. Подпредставление – это представление, созданное на базе другого представления. Вы можете создать «суперпредставление», которое объединяет четыре таблицы, а затем ввести десяток подпредставлений, которые выбирают некоторые колонки, не

> «БОССУ МОЖЕТ НЕ ПОНРАВИТЬСЯ АТОМАРНОСТЬ.»

заставляя вас писать сложные запросы. Синтаксис определения подпредставления ничем не отличается от создания представления на базе обычной таблицы, так что обойдемся без примера. Следуя моему совету, будьте предельно бдительны: легко создать суперпредставление, работающее значительное время, и подпредставления, которым нужна лишь ничтожная часть его данных. Это очень неэкономный подход к использованию ресурсов, так что создавайте подпредставление только если оно содержит столько же таблиц, что и суперпредставление.

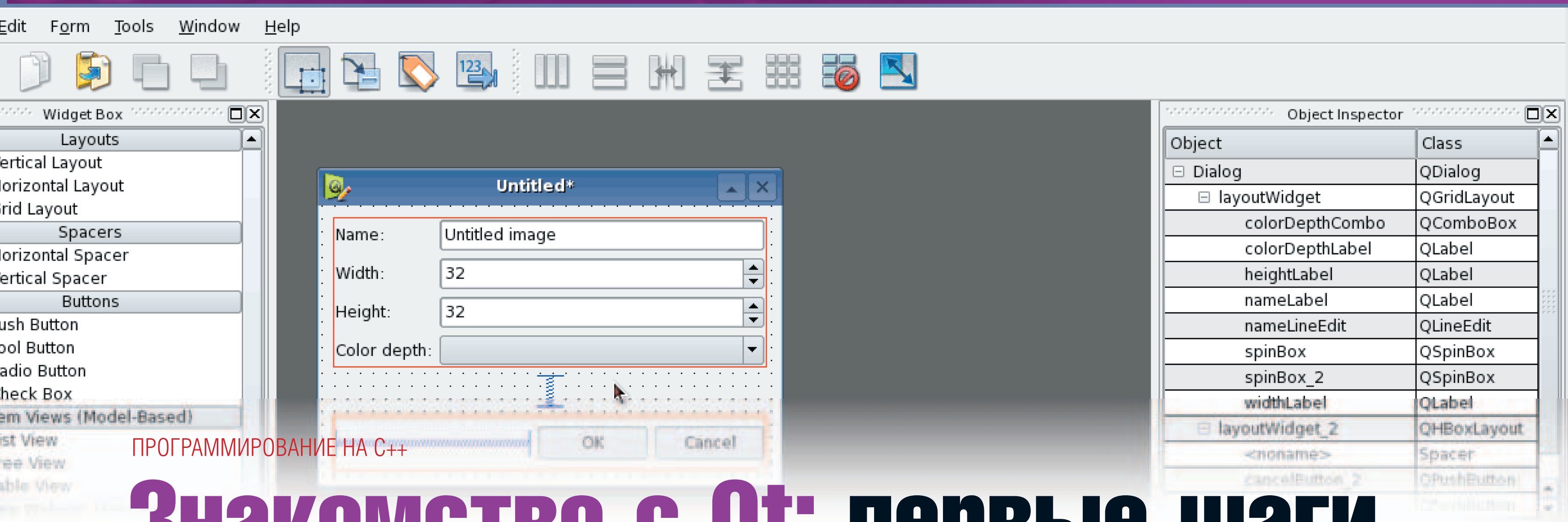
Наконец, если рассматривать представления как функции, вы выходите на совершенно новый уровень контроля доступа к данным. Если для получения необходимой информации используются только представления, вы можете изменить их в любое время и добавить новую колонку или подправить алгоритм. Все пользователи увидят результат нового запроса – без какой-либо правки кода.

Таким образом, представления сокращают время разработки, создают меньше ошибок, добавляют гибкости и помогают убираться по воскресеньям. Ну, насчет уборки я, конечно, загнул, но согласитесь: представления – это круто, и нужно быть сумасшедшим, чтобы не использовать их в своей работе. LXF

СОВЕТЫ

- Если какие-то поля в вашей таблице вычисляются триггером, зачем позволять пользователю устанавливать их значения напрямую? Создайте представления, исключите их из него и пусть триггер делает свое дело.
- Кстати о триггерах – вы не можете использовать их с представлениями, поскольку в них на самом деле не хранятся данные. Устанавливайте триггер на родительскую таблицу.
- И таблицы, и представления принадлежат базе данных, а значит, вы не можете давать им одинаковые имена.
- Чтобы изменить запрос, соответствующий представлению, наберите ту же конструкцию, что и для его создания, но замените CREATE VIEW на CREATE OR REPLACE VIEW.
- Чтобы удалить представление, наберите:

```
DROP VIEW your_view_name;
```



ПРОГРАММИРОВАНИЕ НА C++

Знакомство с Qt: первые шаги

ЧАСТЬ 1 Добро пожаловать в мир графических интерфейсов Linux! **Андрей Боровский** начинает серию учебников для желающих научиться разрабатывать собственные KDE-приложения.

Нет ничего удивительного в том, что многие серьезные приложения для Linux до сих пор обладают только текстовым интерфейсом. Текстовый терминал имеет свои преимущества и даже свою эстетику. Тем не менее, всякий программист, изучающий разработку приложений для Linux, рано или поздно сталкивается с необходимостью освоить программирование графических интерфейсов. Средств разработки графических интерфейсов для приложений Linux существует много, но бесспорными лидерами являются два – Qt/KDE и GTK. Именно с помощью этих средств построены две самые популярные графические среды Linux – KDE и GNOME. Наше внимание будет сосредоточено на разработке приложений с помощью простой объектно-ориентированной системы Qt/KDE.

Система Qt/KDE, как следует из ее названия, состоит из двух слоев – базовой библиотеки классов C++ *Qt toolkit*, разработанной норвежской компанией Trolltech и «надстройки» KDE. К достоинствам *Qt toolkit* следует отнести тщательно проработанную, простую для понимания структуру классов и великолепную документацию. Среда KDE, в свою очередь, развивает традиции, заложенные разработчиками *Qt toolkit*. Изучение разработки приложений для Qt/KDE мы начнем именно с *Qt toolkit*.

Наша первая программа

Что же представляет собой *Qt* с точки зрения программиста? Прежде всего – это библиотека, насчитывающая более четырехсот классов C++. Классы *Qt* реализуют различные элементы графического интерфейса пользователя, интерфейсы взаимодействия с базами данных и некоторые часто используемые структуры данных. Помимо этого *Qt* включает визуальный редактор пользовательского интерфейса *Qt Designer*, набор инструментов для интернационализации приложений *Qt Linguist* и браузер справочной документации *Qt Assistant*. *Qt Assistant* составляет основу справочной системы самой *Qt* и может использоваться разработчиками для реализации интерактивной справки в их собственных приложениях. *Qt* – кросс-платформенная библиотека, она реализована на платформах Windows, Mac OS X, X11 (Linux, BSD, Solaris, HP-UX, IRIX, AIX и другие). На платформах Windows и X11 библиотека напрямую обращается к низкоуровневым API (GDI для Windows и X API для X11), благодаря чему обеспечивается максимально возможное для быстродействие.

Хотя на момент написания статьи компания *Trolltech* уже выпустила *Qt* 4, все примеры статьи написаны с использованием *Qt* 3.3.5, так как именно на этой версии основана новейшая на сегодняшний день версия KDE – 3.5.1. Хотя простейшие примеры из этой статьи можно скомпилировать на всех версиях *Qt*, для компиляции более сложных примеров из последующих статей цикла может понадобиться версия 3.3.5 или выше. Кроме пакетов основной библиотеки в системе должны быть установлены пакеты разработчика (*qt-devel*, *kde-devel* или подобные). К версии компилятора никаких особых требований нет. В системе также должны быть установлены инструменты сборки и конфигурации исходных текстов: **make**, **autoconf** и **automake**. Для удобства работы с *Qt* следует установить переменные окружения. В переменную PATH добавим путь к утилитам *Qt*, например:

```
PATH=/usr/local/Trolltech/Qt-4.1.1/bin:$PATH
export PATH
```

Для того, чтобы утилиты и make-файлы *Qt* выполнялись успешно, необходимо создать переменную *QTDIR*:

```
export QTDIR=/usr/lib/qt3
```

Структура библиотеки *Qt* довольно проста и напоминает структуру библиотек MFC и VCL. Родоначальником большинства классов *Qt* является QObject, а все визуальные элементы управления происходят от одного из его потомков – класса QWidget. Классы объявляются в заголовочных файлах *Qt*, имена которых соответствуют именам классов. Например, класс QPushButton объявлен в файле **QPushButton.h**. Справочная система *Qt*, помимо прочего, предоставляет подробную информацию обо всех классах библиотеки. Доступ к справочной системе можно получить двумя способами: открыть копию индекса справочной системы на локальном жестком диске (это можно сделать с помощью приложения *Qt Help Assistant*, показанном на *рис.1*, а можно – с помощью web-браузера) или найти нужный раздел справки на специальном сайте **doc.trolltech.com**.

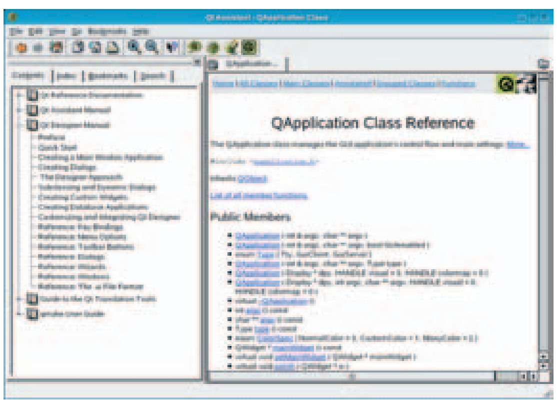


Рисунок 1. Qt Help Assistant

Для того, чтобы познакомиться с классами *Qt* на практике, напишем простую программу:

```
#include <QApplication.h>
```

```
#include <QLabel.h>
int main(int argc, char *argv[])
{
 QApplication MyApp(argc, argv);
 QLabel Label(NULL);
 Label.setFrameStyle(QFrame::Panel | QFrame::Sunken);
 Label.setText(«Hello, brave new Qt world!»);
 MyApp.setMainWidget(&Label);
 Label.show();
 return MyApp.exec();
}
```

Эта программа, своего рода «*Hello world!*» для *Qt,* включает в себя все основные элементы приложения. Рассмотрим текст программы по порядку. В файле **QApplication.h** объявлен класс QApplication, который играет центральную роль в работе приложения *Qt*. Файл **QLabel.h** декларирует класс QLabel, соответствующий элементу интерфейса «панель со статическим текстом» (также известному как «метка»).

Как и в любой Linux-программе, написанной на C++, точкой входа в *Qt*-программе является функция **main**. В этой функции мы, прежде всего, создаем объект класса QApplication. У класса QApplication несколько конструкторов, мы выбираем из них конструктор с двумя параметрами, аналогичными параметрам функции **main**. Получив переменные argc и argv, конструктор извлекает из них ключи, которые относятся к конфигурации приложения X Window (дисплей для вывода, геометрия окна), если, конечно, таковые имеются. Таким образом, используя объект класса QApplication, мы «бесплатно» получаем приложение, которое ведет себя в соответствии со стандартами X Window! Далее мы создаем объект класса QLabel, который будет главным (и единственным) визуальным элементом нашего приложения. В заголовке выбранного нами конструктора класса QLabel определен один параметр типа QWidget*. В этом параметре конструктору передается указатель на класс, содержащий родительский визуальный элемент (родительский в иерархии визуальных элементов окна, а не в иерархии классов). Аналогичный конструктор имеют и другие визуальные элементы *Qt*: кнопки, поля ввода и т.д. У главного визуального элемента приложения родителя нет, поэтому мы передаем NULL конструктору класса. Метод setFrameStyle, унаследованный классом QLabel от его предка – класса QFrame, позволяет указать некоторые параметры внешнего вида панели. Флаги, передаваемые методу setFrameStyle, являются элементами класса QFrame. Метод setText задает текст, который отображается элементом QLabel. Далее, с помощью метода setMainWidget, мы указываем объекту MyApp, что Label является главным визуальным элементом приложения. Метод show делает объект Label видимым. Последнее, что мы делаем в нашей программе – запускаем цикл обработки сообщений с помощью метода exec объекта MyApp. Цикл будет выполняться до тех пор, пока приложение не получит команду завершить работу.

Теперь можно приступить к сборке приложения. Сохраняем исходный текст в отдельной директории в файле с именем **app1.cpp**. Наша следующая задача, – создать проект приложения. Для генерации проекта используется утилита *qmake*. Переходим в директорию, в которой сохранен файл **app1.cpp** и даем команду

```
qmake -project
```

В результате в директории должен появиться файл проекта **app1.pro**. Теперь, чтобы получить make-файл проекта, достаточно скомандовать

```
qmake
```

После этого собираем наше приложение командой **make**. Запустив на исполнение двоичный файл **app1**, мы увидим простое окно с текстовой панелью (*рис. 2*).

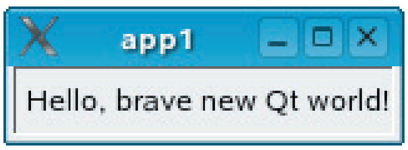


Рисунок 2. «Здравствуй, *Qt*!»

Обработка событий

Как и многие другие графические системы, *Qt toolkit* базируется на событийно-управляемой архитектуре. Когда в графической системе, например X Window, происходит нечто, связанное с одним из окон приложения (щелчок мышью, нажатие на клавишу, сокрытие окном другого приложения – то есть необходимость перерисовки), этому окну посылается сообщение. *Qt toolkit* преобразует сообщение графической системы в событие *Qt,* создает объект класса события и вызывает метод-обработчик события, которому в качестве параметра передается созданный объект. Обработчик события, связанного с окном, – это виртуальный метод объекта, отвечающего за данное окно. Методы-обработчики, как правило, принимают один параметр, имеющий тип «класс события». Все классы событий *Qt* являются потомками класса QEvent. Класс QWidget включает в себя обработчики практически всех событий, которые может обрабатывать визуальный элемент, однако большинство этих методов – заглушки, не выполняющие никаких действий. В классах-потомках QWidget переопределяются методы-обработчики событий, актуальных для соответствующих классов. Для того, чтобы задать собственный обработчик события для какого-либо класса *Qt,* необходимо создать потомок этого класса и переопределить в нем соответствующий метод базового класса.

Рассмотрим описанный механизм на примере создания обработчика события mousePressEvent для потомка класса QLabel. Событие mousePressEvent генерируется в результате нажатия кнопки мыши, если в этот момент указатель мыши находится в области окна. Событие mousePressEvent обрабатывается методом mousePressEvent, объявленным в базовом классе QWidget, где этот метод является заглушкой. Метод mousePressEvent не переопределяется классах QFrame и QLabel, поскольку соответствующие визуальные элементы обычно не реагируют на нажатие кнопки мыши. В следующем примере мы создаем производный класс от QLabel, в котором переопределяем метод mousePressEvent.

```
class QMyLabel: public QLabel
{
		public:
		QMyLabel( QWidget * parent, const char * name = 0, WFlags
f = 0)
				:QLabel(parent, name, f)
		{
		}
				protected:
				void mousePressEvent ( QMouseEvent * e )
		{
				QString S;
				S = QString(«x= %1, y= %2»).arg(e->x()).arg(e-
>y());
				this->setText(S);
		}
};
```

У обработчика mousePressEvent один параметр типа QMouseEvent*. QMouseEvent – это класс, соответствующий событиям, связанным с мышью. Объект этого класса содержит информацию о событии – координаты указателя мыши в момент события, а также

>>

QT И OPEN SOURCE

Лежащий в основе KDE инструментарий *Qt toolkit* распространяется на условиях двойного (сейчас даже можно сказать – тройного) лицензирования. Разработчики Open Source могут распространять свои приложения на условиях открытой Q Public License (QPL) или GPL. Лицензия QPL, одобренная организацией Open Source Initiative позволяет распространять *Qt toolkit*, а также программное обеспечение, созданное с его помощью, в форме исходных текстов и двоичных модулей на бесплатной основе. Модификации исходного кода должны распространяться отдельно в виде патчей на тех же условиях. Разработчики коммерческих продуктов должны приобрести коммерческую лицензию, которая позволяет не открывать исходные тексты, а также предоставляет в распоряжение разработчика дополнительные возможности коммерческих расширений *Qt*.

состояние кнопок мыши и специальных клавиш. Используя информацию объекта e, обработчик формирует строку, содержащую координаты указателя мыши в момент возникновения события и выводит эту строку на панель с помощью метода setText., используя перегруженный метод arg класса QString (строк). Для подстановки значений в строку S применяется метод arg класса QString – привычный C-программистам snprintf в *Qt* используется редко. Теперь заменим в исходной программе класс *QLabel* классом QMyLabel:

```
int main(int argc, char *argv[])
{
        QApplication MyApp(argc, argv);
        QMyLabel Label(NULL);
        Label.setFrameStyle(QFrame::Panel | QFrame::Sunken);
        Label.resize(200, 100);
        Label.setFont( QFont( «Courier», 14, QFont::Bold ) );
        MyApp.setMainWidget(&Label);
        Label.show();
 return MyApp.exec();
}
```

Мы получили программу, отображающую в своем окне координаты щелчка мыши:

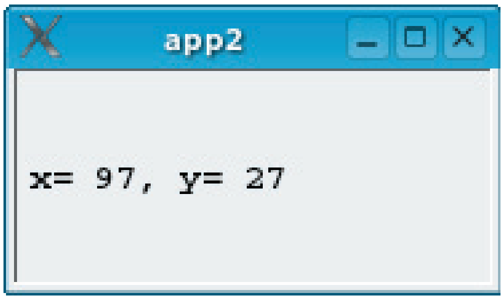


Рисунок 3. «Умная» метка

Как узнать, какие события может обрабатывать тот или иной класс *Qt*? Для ответа на этот вопрос можно составить список методов-обработчиков событий для данного класса и его предков с помощью справочной системы *Qt*. Методы-обработчики можно отличить от других методов по их именам, которые оканчиваются на «Event».

Сигналы и слоты

В приложениях, построенных на основе иерархии объектов, часто бывает необходимо, чтобы в ответ на событие, связанное с одним из объектов, вызывался метод другого объекта. Допустим, у нас есть объект Button, представляющий кнопку. Нажатие кнопки должно вызвать изменения в другом объекте, например, удалять текст, выводимый объектом Label класса QLabel. Для решения этой задачи можно конечно, переопределить событие mousePressEvent, в потомке класса QPushButton (кнопка), как мы это делали раньше, но такой подход оказывается весьма неудобным: нам придется создать для нашей кнопки Button новый класс-потомок QPushButton, учитывающий логику взаимодействия с объектом Label. Может случиться так, что для каждой кнопки в нашей программе нам понадобится создать собственный класс C++. В общем случае, если мы хотим, чтобы класс A реагировал на события, связанные с классом B, нам придется создавать производные классы от A и от B. Излишне говорить, что описанный метод весьма громоздок и неудобен. Все интегрированные системы разработки на C++ пытаются так или иначе упростить решение этой проблемы. Одним из альтернативных решений является метод обратных вызовов (callbacks). При использовании метода обратных вызовов объекту источнику события (например, Button), передается адрес метода объекта обработчика (в нашем примере – адрес метода clear объекта Label). Конечно, обратные вызовы можно реализовать в явном виде средствами стандартного C++, но такое решение будет почти столь же трудоемким, как и описанное выше. *Qt* library упрощает взаимодействие между объектами при помощи основанного на методе обратного вызова механизма сигналов и слотов. Объект *Qt* генерирует (в терминологии *Qt* – «эмитирует») сигнал при изменении своего состояния. Если данный сигнал объекта связан со слотом другого (или того же самого) объекта, выполняется метод, соответствующий слоту. Связь сигнал-слот можно рассматривать как однонаправленную линию коммуникации между двумя объектами.

С точки зрения разработчика слоты являются обычными методами классов Qt. Если сигнал несет какие-либо данные, эти данные могут быть переданы слоту через его параметры. Существует возможность связывать один сигнал с несколькими слотами и один слот с несколькими сигналами. Это означает, например, что событию может быть сопоставлено несколько обработчиков, являющихся методами разных объектов, и в ответ на событие будут вызываться все назначенные ему обработчики. Следует помнить, что сигналы и слоты, – это не новые элементы языка программирования, а всего лишь удобный способ описания методов классов. Синтаксис сигналов и слотов реализован в виде макросов, которые обрабатывает специальный препроцессор *Qt* – moc (Meta Object Compiler).

Для того, чтобы проверить описанные концепции на практике, напишем еще одну программу:

```
#include <QApplication.h>
#include <qvbox.h>
#include <QLabel.h>
#include <qlineedit.h>
#include <QPushButton.h>
int main(int argc, char *argv[])
{
        QApplication MyApp(argc, argv);
        QVBox MainBox;
        MainBox.resize( 150, 110 );
        QLabel Label(&MainBox);
        QLineEdit Edit(&MainBox);
        QPushButton Button(«Clear», &MainBox);
        Label.setFrameStyle(QFrame::Panel | QFrame::Raised);
        QObject::connect(&Edit, SIGNAL(textChanged(const
QString&)), &Label, SLOT(setText(const QString& )));
        QObject::connect(&Button, SIGNAL(clicked()), &Label,
SLOT(clear()));
        QObject::connect(&Button, SIGNAL(clicked()), &Edit,
SLOT(clear()));
        MyApp.setMainWidget(&MainBox);
        MainBox.show();
 return MyApp.exec();
}
```

Эта программа содержит много новых элементов, поэтому рассмотрим их по порядку. Класс QVBox представляет собой панель, способную упорядочивать расположение дочерних визуальных элементов по вертикали. В этом смысле *Qt* (и *Gtk*) значительно отличается от инстру-

ИСТОРИЯ ОТКРЫТОГО ЛИЦЕНЗИРОВАНИЯ QT

Компания Trolltech далеко не сразу согласилась на открытое лицензирование своего продукта. Использовавшаяся в начале лицензия *FreeQt* не позволяла свободно распространять модификации исходных текстов *Qt toolkit*. Когда основанная на *Qt* графическая оболочка KDE начала завоевывать популярность на платформе Linux, у многих сторонников Open Source сложилось убеждение, что лицензия *FreeQt* ставит под угрозу дальнейшую судьбу KDE как открытого проекта. Участники движения Open Source опасались, что в определенный момент ведущая графическая среда для Linux может превратиться в закрытый коммерческий продукт. Оппозиция сообщества Open Source привела к постепенному увеличению открытости главного продукта компании Trolltech. В результате компромисса была принята новая лицензия Q Public License. Специальная договоренность гарантировала, что условия QPL не станут более жесткими даже в том случае, если компания разработчик перейдет в третьи руки. Организация KDE Free Qt Foundation должна была обеспечить переход *Qt toolkit* под действие лицензии BSD License в случае, если в течении 12 месяцев не будет выпущено открытой версии Qt. Тем не менее, разработчики, не очень доверявшие изменениям лицензионной политики, основали проект Harmony, призванный создать полностью открытую библиотеку, эмулирующую *Qt*. Проект Harmony был закрыт после того, как Trolltech выпустила Qt Open Source Edition на условиях GPL.

ментариев, доступных Windows-программистам. В VCL, MFC и прочих применяется модель абсолютного позиционирования, когда положение элемента управления на форме, а также его размеры задаются координатами в пикселях. При изменении размеров окна программа сама должна пересчитать координаты его дочерних элементов. В мире Unix еще со времен Tk преобладает другой подход – менеджеры компоновки (layout manager). Разработчик просто указывает, как нужно расположить элементы на форме (вертикально, горизонтально, как в таблице и т.п.), а компоновщик берет на себя всю работу по вычислению координат и автоматически обновляет их при изменении размеров окна. В случае с QVBox, дочерние элементы располагаются друг под другом в порядке их объявления. Объект MainBox класса QVBox является главным визуальным элементом нашего приложения. Метод resize изменяет размеры главного элемента. В конструкторе QLabel мы указываем MainBox в качестве родительского элемента. Класс QLineEdit – это строка ввода, класс QPushButton – кнопка. Первый аргумент конструктора QPushButton – текст кнопки, второй аргумент – ссылка на родительский визуальный элемент. Таким образом, в окне программы будут расположены три элемента – текстовая панель, строка ввода и кнопка. Высота кнопки и строки ввода фиксирована, а текстовая панель займет всю оставшуюся область окна.

Далее в программе следуют три конструкции связывания сигналов и слотов. Связывание выполняет статический метод connect класса QObject. У метода connect 4 аргумента. Первый аргумент – указатель на объект-источник сигнала, второй аргумент – макрос SIGNAL, которому передается описание сигнала, третий аргумент – указатель на объект-владелец слота и последний аргумент – макрос SLOT, которому передается описание слота. Конструкция

```
QObject::connect(&Edit, SIGNAL(textChanged(const QString&)), &Label,
SLOT(setText(const QString& )));
```

означает, что сигнал textChanged объекта Edit связывается со слотом setText объекта Label. Сигнал textChanged эмитируется объектом класса QLineEdit всякий раз, когда меняется содержимое строки ввода, а слот setText класса QLabel аналогичен уже знакомому нам одноименному методу (далее мы увидем, что слоты представляют собой не что иное как методы классов C++, так что слот можно не только связывать с сигналом, но и вызывать напрямую, как обычный метод объекта). У сигнала textChanged и слота setText одинаковый набор параметров (один параметр типа «ссылка на объект QString»), и эти параметры мы указываем в описании сигнала и слота. Указывать параметры необходимо для того, чтобы слот setText получал строку, передаваемую сигналом textChanged. Мы можем не указывать параметры или вообще связать сигнал и слот с разными списками параметров, но в этом случае слот не получит данных, а будет использовать значения своих параметров, заданные по умолчанию. Возможность связывать сигналы и слоты с разными списками параметров очень удобна, но она может стать источником ошибок, которые трудно будет обнаружить. Связывание сигнала textChanged и слота setText приведет к тому, что любое изменение в строке ввода будет тут же отображаться на текстовой панели.

Далее мы связываем сигнал clicked объекта Button со слотом clear объекта Label. Сигнал clicked эмитируется объектами класса QPushButton в ответ на щелчок по кнопке. Слот clear удаляет текст с панели. Связывание этих сигнала и слота приводит к тому, что нажатие кнопки Button будет удалять текст с панели Label. Следующая строка программы выполняет связывание того же сигнала **Button.clicked** с слотом clear объекта Edit, удаляющим текст из строки ввода. Использование сигнала clicked настолько удобнее по сравнению с обработкой события clicked, что разработчики *Qt* даже не вели в состав класса QPushButton такое событие. Мы назначаем объект MainBox главным визуальным элементом приложения. Мы должны сделать MainBox видимым с помощью метода show, но для дочерних объектов этого делать не нужно, поскольку MainBox сам позаботится о них.

Программа собирается той же последовательностью вызовов, что и предыдущая:

```
qmake –project
qmake
make
```

ПОЛЕЗНЫЕ ССЫЛКИ

Если вы заинтересовались *Qt* и хотите узнать больше об этом инструментарии, советуем вам обратить внимание на книгу Жасмин Бланшетт (Jasmin Blanchette) и Марка Сомеерфильда (Mark Summerfield) «C++ GUI Programming with Qt 3». Эта книга написана сотрудниками компании Trolltech и выходит в серии «Bruce Perens Open Source Series». Книга переведена на русский язык и выпущена издательством «Кудиц-Образ». Если вам не удалось найти ее в ближайшем книжном магазине, вы можете загрузить бесплатную английскую PDF-версию с сайта phptr.com/perens. Перевод на русский язык доступен для чтения онлайн: **http://www.linuxcenter.ru/lib/books/qt3/**. Ну и не забывайте о восхитительной справочной документации *Qt*!

Запустив скомпилированную программу на выполнение, мы увидим вполне интерактивное приложение, правильно реагирующее на изменение размеров окна и занимающее всего 13 строк кода. Неплохо для начала!

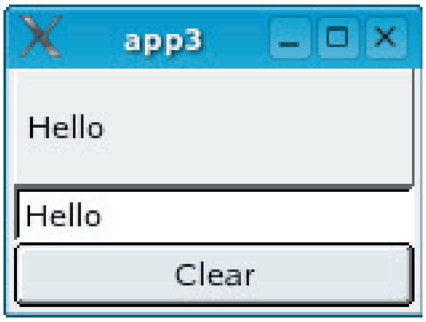
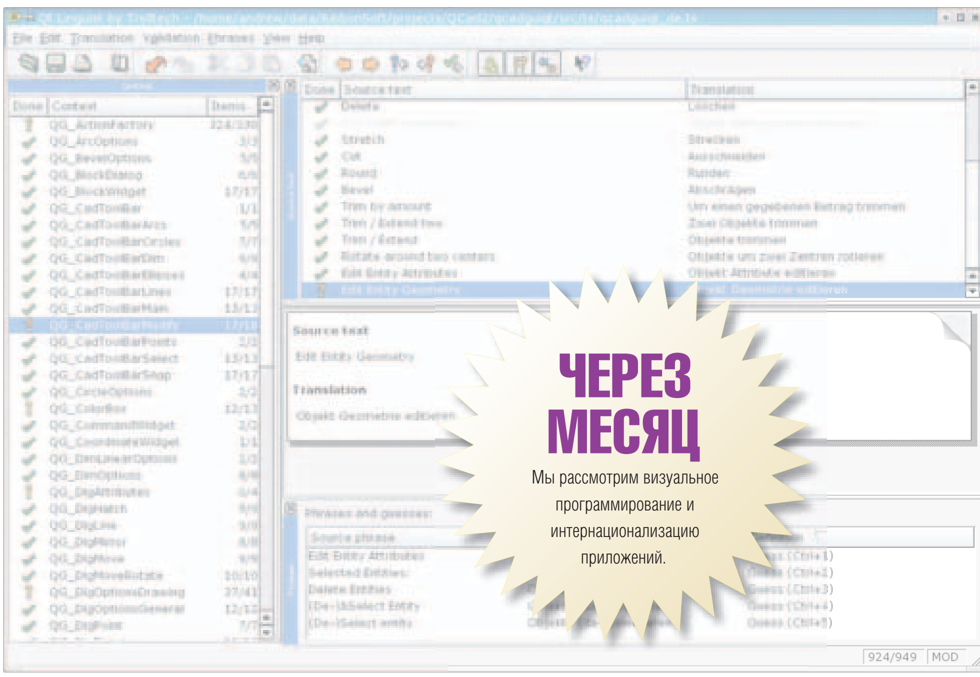


Рисунок 4. Слоты и сигналы делают программирование простым и понятным.

Соединение сигнал-слот можно разорвать с помощью статического метода QObject::disconnect. Список параметров у этого метода такой же, как и у метода connect.

Мы рассмотрели лишь основы программирования *Qt*, однако этих сведений достаточно для написания простых программ с графическим интерфейсом. Конечно, мы не собираемся останавливаться на достигнутом. Продолжение следует! LXF



ЧЕРЕЗ МЕСЯЦ

Мы рассмотрим визуальное программирование и интернационализацию приложений.

ТЕКСТОВЫЙ РЕДАКТОР

Почему – VIM?

Давнее противостояние продолжается. Адвокат Vim **Владимир Попов** поднимается на трибуну, чтобы дать достойный ответ оппоненту...

€ 15 – за что?!

Недавно увидел предложение: plugin *VIM* для его интеграции в Eclipse. Недорого – всего 15 евро. Если кто-то не знает, Eclipse – это, помимо прочего, кроссплатформенная интегрированная среда разработки, иначе говоря – IDE. Из этого следует, что свой редактор в Eclipse обязательно входит. Кому же может потребоваться интеграция Eclipse с *VIM*, да ещё и «не даром»? Кому-то, как выясняется, требуется. Что же это за «динозавр» такой, если его качества кажутся не лишними для современной IDE? Или это просто «дань уважения традиции»? Без выяснения, что же такое *VIM*, ответить на этот вопрос не удастся.

Тридцать лет – это возраст...

Все нынешние потомки *VI* (а их насчитывается полтора десятка для всех наиболее популярных ОС) имеют общего предка – VIsual editor Билла Джоя (Bill Joy), написанный в 1976 году. История, типичная для времени становления UNIX: несколько прототипов (**ed/em/ex**), несколько оригинальных идей, коллективное творчество... и удачный продукт, если повезёт. По мнению самого Джоя, от EMACS его редактор изначально отличали, прежде всего:

- мультирежимность (mode-based editing);
- не-программируемость;
- цена (EMACS в те времена стоил несколько сот долларов).

Иными словами: никаких особенных претензий. Доступность редактора, по мнению Джоя, была единственным основанием для его включения в качестве базового во многие UNIX-системы. Глядя из 2006-го, поневоле задумаешься: а ведь, похоже, будущие принципы Open Source «угадывались» уже в далёкие 70-е...

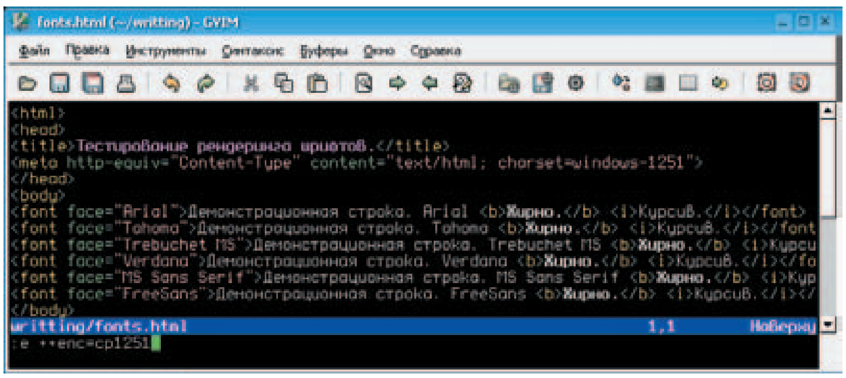
Так изначальная простота и доступность обеспечили распространение *VI*. Без сомнения, их можно занести в актив программы и сейчас. Открытость гарантировала наращиваемость (результат – почти четыре мегабайта исходных текстов *VI* IMproved Брэма Моолинаара (Bram Moolinaar)), а простота приводит к тому, что если нужно выбрать редактор для встроенной системы (см. *busybox*), то это, вполне вероятно, будет *vi*. Похоже, тридцать лет прошли не напрасно.

UNIX-ово племя

Ориентировочно, только пользователей вышеупомянутого *VIM* насчитывается более миллиона и, разумеется, это не только пользователи UNIX. Но то, что именно последним он ближе всего, сомнений не вызывает. Для этого, как мне кажется, существуют две предпосылки.

Во-первых, сам *VIM* следует принципам UNIX, насколько это только возможно для редактора. Последовательное воплощение принципа KISS (Keep It Simple, Stupid!) не всегда реализуемо (не так часто *VIM* становится звеном цепочек обработки и прочих pipe-ов), но стремление следовать «заветам» налицо. Во всяком случае, перенаправление вывода команды в редактируемый текст и, наоборот, использование части текста в качестве команды, широкое использование «конвейерной» обработки и т.п. неизменно радуют сердце «юниксоида». В *VIM* нет проверки правописания, но есть всё необходимое для использования любого из наличествующих в системе spellchecker-ов. И так со всеми сопутствующими редактированию процессами. Потребовалось редактирование двоичных файлов – добавим простейший конвертор, а редактор текста пусть остается редактором текста.

Во-вторых, UNIX, для которого текстовые файлы – «плоть от плоти», сам «благоволит» к *VI* и его потомкам. Многие программы вызывают *VI* как «штатное» средство редактирования, «горячие клавиши» *VI* часто используются в том же качестве другими программами, существует стиль редактирования *VI* и даже *bash*, по умолчанию предлагающий стиль редактирования строки *emacs*, допускает переназначение переменной editing-mode на 'vi'. Общий подход прослеживается при использовании регулярных выражений и т.д. и т.п. Возможно, Ctrl+F для поиска выглядит и более логично (если под F подразумевать Find, то есть «находить»), но привычное для *VI* нажатие / (слэш), как ни странно, даст желаемый результат что при просмотре man-страниц, что в *links*, что в *mcedit*.



GVim **во всей своей красе.**

Чуть подробнее

VIM последовательно (иногда даже навязчиво) придерживается программистской логики: есть объекты (символ, слово, строка, предложение, параграф, блок) и есть действия (позиционирование, копирование, удаление, вставка, замена и т.д). Все действия возможны над всеми объектами от одного до N раз. Откуда следует универсальная команда, подобная следующей:

```
[N раз] { идентификатор действия } [ идентификатор объекта ]
```

Теперь нужно оптимизировать принципы идентификации объектов и ввода команд. Результат, кстати, будет зависеть от пожеланий пользователя и устройств ввода и отображения (напомню: зарождалась эта идеология во времена довольно разнообразных терминалов). Кому-то представляется достаточной система меню, кому-то – «горячие» клавиши, а кто-то предпочтёт и исключающую двусмысленность командную строку. *VIM* это уже не волнует: обеспечив возможность выбора, авторы предоставляю пользователю дальнейшее развитие продукта. Логично, поскольку программа открыта и бесплатна: нет никаких оснований тратить время на её «упаковку», ориентированную на ту или иную целевую группу (имею в виду хорошо известные маркетологам target group).

Подобная «жёсткая» логика в применении к файлам выглядит следующим образом:

- файлы можно создавать, удалять, читать из них, писать в них;
- открытому файлу соответствует буфер;
- окно предоставляет доступ к отдельному буферу.

Разумеется, число открытых файлов (и соответствующих им буферов) ничем не ограничивается.

Разумеется, число окон ограничивается только здравым смыслом и площадью экрана.

Разумеется, число окон никак не связано с числом буферов (открытых файлов).

Разумеется, окна могут делить экран как угодно (по горизонтали, по вертикали).

Разумеется, есть множество команд, выполняющих все возможные (иногда: непонятно зачем) действия над файлами, буферами и окнами. А выбор (меню, «горячая клавиша» или командная строка) – за пользователем.

Создаётся впечатление, что авторы задумывали универсальный инструмент работы с текстом, а как именно будет применять его пользователь – предоставили решать ему самому. Всё многообразие команд не помещается ни меню, ни в набор «горячих клавиш». Можно, конечно, свести все действия к командной строке с двумя режимами: редактирования и командным. Либо – редактируем, либо – вводим команду: просто, определённо и... утомительно. Так появились дополнительные режимы. Вообще-то у *VIM* шесть основных режимов и пять дополнительных – мало не покажется. Но и пугаться особенно не стоит: кроме само собой разумеющихся режимов редактирования («вставки») и командного (скромно именуемого нормальным или обычным), для начала достаточно знать ещё визуальный (перемещение курсора изменяет область выделения) и «замены» (вводимые символы «замещают» содержимое буфера, а не дополняют его).

Переключение между режимами, не очевидное на первых порах, также не вызывает трудностей: одно-два нажатия Esc всегда вернут в нормальный режим, первый Ins переводит в режим «вставки», второй – «замены». v, V или Ctrl+V обеспечивают переход из нормального режима в визуальный, причём выделение будет выполняться посимвольно, по строкам или в виде прямоугольного блока – в зависимости от того, который из перечисленных символов введен.

На этом первое знакомство с *VIM* можно и закончить: не предполагали же вы, что данная статья заменит вам 4 МБ документации в текстовом формате? Справедливости ради отметим, что 3.5 МБ из 4 – это встроенная система помощи (она же – Reference Guide). Для чтения больше подходят оставшиеся 0.5 МБ User's Guide. Но и это не мало – не правда ли?

На этом этапе впору задуматься: если знакомство с *VIM* так трудоёмко, то стоит ли этим заниматься вообще? Ситуация напоминает первое знакомство с Linux: есть желание и (хочется верить) способность разобраться, но чтобы было с чем разбираться, надо сначала что-то инсталлировать. А вот на это-то у «неофита» то ли знаний, то ли терпения может и не хватить... Что ни говори, а знакомиться с уже работающей «игрушкой» намного интереснее, чем читать документацию, постоянно мысленно возвращаясь к одному и тому же вопросу: зачем?

Можно надеяться, что с современными инсталляциями *VIM* так не случится: и учебничек на русском языке к нему прилагается (запускается просто: **vimtutor**, проверьте только, чтобы **/usr/share/vim/vimNM/tutor/tutor** указывал на **tutor.ru**), и сам он по-русски общается с пользователем довольно сносно, а если хранитель дистрибутива позаботился ещё и о более-менее полных вариантах **vimrc_example.vim** (из которого следует немедля сделать **~/.vimrc**) и **menu.vim**, то «игрушку» можно признать вполне работоспособной. А уж насколько она окажется полезной – зависит только от пользователя.

Осталось выяснить: нужно ли это вам вообще? Если с текстом случается работать лишь изредка, то скажем прямо: нет. *VIM* включает в себя в качестве расширений и игрушки (всем известный *Tetris* и *Башню Ханоя*, например), но только из-за этого связываться с ним не стоит. Общее знакомство может пригодиться в тот несчастливый день, когда вы окажетесь один на один с чёрным экраном консоли и каким-нибудь rescue-cd (последний других средств редактирования файлов может и не иметь), если, конечно, вспомните к тому моменту хотя бы команды открытия/закрытия файлов (именно на этот случай мы прилагаем во врезке краткий список «тривиальных» команд, доступных даже в самых усечённых реализациях *VI*).

Так или иначе, но даже поверив «на слово», что знакомство с *VIM* будет не таким уж утомительным, хотелось бы убедиться, что оно может быть полезным. Ниже следует список возможностей, которые, по моему скромному мнению, разумеется, могут заинтересовать читателя.

Что особенного-то?

Кроме упомянутых выше неограниченного числа буферов (открытых файлов) и произвольного числа окон, хочется отметить возможности *VIM*, несколько выходящие за пределы стандартных:

- *VIM* работает в текстовом режиме на большинстве терминалов, но для любителей есть и графический интерфейс (то есть меню и поддержка мыши). Излишняя в эру IBM PC поддержка терминалов оказывается очень кстати при открытии сессии на удалённом компьютере. Согласитесь, что просмотр журналов или редактирование конфигурационных файлов сервера, у которого, может, и терминала-то нет, удобнее выполнять с помощью привычного редактора;
- *VIM* выполняет автодополнение строк, слов, имён файлов и команд. В зависимости от контекста (режима) и формы запроса (определяется последовательностью «горячих клавиш») в качестве вариантов будут предложены слова или строки из текущего и подключаемых файлов, так называемых 'dictionary' и 'thesaurus', имена открытых файлов, описанные определения и макросы либо допустимые элементы командной строки;
- *VIM* поддерживает редактирование «справа налево» (арабский, персидский языки, иврит) и многобайтовые символы (китайский, японский, корейский языки), не говоря уже обо всех 8-битных кодировках кириллицы плюс, разумеется, UTF-8 и Unicode;
- *VIM* умеет выполнять автокоманды: выполнение определённых действий в зависимости, например, от типа файлов;
- помимо обычных в наше время «undo» и «redo», *VIM* сохраняет «историю» команд и поисков, так что их можно повторно использовать, а, при желании, и редактировать;
- количественный префикс (см. выше «универсальную команду») позволяет указывать количество повторений для большинства команд;
- *VIM* сохраняет информацию о редактировании в текстовом файле **~/.viminfo**. Эта информация позволяет не только восстановить сеанс

>>

VI-МИНИМУМ

Ещё раз напомним, что главное при работе в **VI** – контролировать, в каком режиме находится редактор. Нормальному режиму соответствует пустая нижняя строка экрана. Все остальные режимы индицируются надписью в левом нижнем углу. Esc всегда возвращает в нормальный режим, Ins переводит в режим «вставки»/»замены» (экранное редактирование), v переводит в визуальный режим (режим выделений).

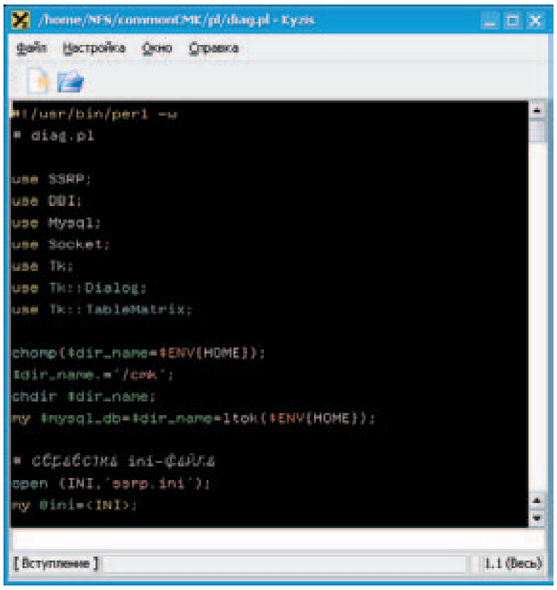
В нормальном режиме возможны:

- :e **file** – открытие существующего или создание нового файла;
- :w **[file]** – сохранение буфера в «свой» или новый файл. :w! – невзирая на защиту записи;
- Ctrl-G – дать полную информацию об открытом файле и положении курсора в его буфере;
- :q – закрытие окна (для последнего окна – выход из редактора). :q! – невзирая на наличие несохранённых буферов;
- :!**command** – выполнить внешнюю команду;
- :bn – переключиться на следующий буфер;
- Ctrl-W**,** s – разделить окно пополам по горизонтали, Ctrl-Wv – по вертикали;
- Ctrl-W**,** w – перейти в следующее окно;
- d – удалить символ с помещением в регистр обмена, dd – строку;
- y – копировать в регистр обмена символ, yy – строку;
- [p**,**]p – вставить из регистра после или перед курсором;
- :r **file** – вставить содержимое файла;
- **/string** – искать string (строка или регулярное выражение) от курсора и ниже;
- **?string** – искать string выше от курсора;
- n или N – продолжить поиск ниже или выше по тексту;
- **%s/string_old/string_new** – замена во всём буфере;
- * – найти следующее вхождение слова под курсором, # – предыдущее;
- % – найти ответную скобку;
- . – повторить последнюю операцию;
- **[n]**G – перейти на строку n, в отсутствие n – в конец файла;
-) или (следующее или предыдущее предложение, для **}/{** – параграф, для **]]/[[** – секция, функция;
- mx – пометить текущую позицию буквой x;
- **`x** – перейти к метке **x**;
- ~ – изменить регистр символа под курсором;
- u – «undo», U – восстановить всю строку;
- Ctrl-R – отмена undo.

Список весьма неполон и, наверное, субъективен, но что поделаешь – полное описание только способов позиционирования будет иметь объём больший, чем весь наш «минимум». А ведь есть ещё команды удаления, замены, информационные...

В визуальном режиме d и y в качестве объекта подразумевают выделение, как и меняющие регистр u**/**U.

О режиме «вставки» (поскольку мы говорим практически исключительно об IBM PC) достаточно сказать, что работают все клавиши позиционирования курсора. Хоть это и не вся правда, но напомним: данный минимум предназначен только для того, чтобы справиться со «спартанскими» вариантами **VI** из состава *busybox* или *crunchbox*.



***Yzis* мал, да удал.**

<< редактирования после краха, но и иметь своеобразную «заготовку» среды редактирования – нечто вроде «проекта» со своим списком открытых файлов, шаблонов и макрокоманд;

- гордость последних версий *MS Office*, многостраничный буфер обмена, присутствует в *VIM* в форме регистров – нумерованных (для **cut/copy/paste**) и именованных, обращение к которым задаётся в явном виде. Содержимое именованных регистров может, кроме того, накапливаться (нечто вроде Memory+ у калькулятора), а также использоваться, как команда. Есть еще несколько специальных регистров (последняя вставка, команда, файл, буфер обмена *X Window*);
- для облегчения поиска *VIM* может расставлять в тексте собственные метки, а может использовать систему «тегов» для индексирования текста. Эдакий «markup language» в миниатюре;
- *VIM* поддерживает так называемые «складки» (foldings): часть текста, определяемая вручную, в зависимости от отступа либо в соответствии с синтаксисом или специальными маркерами как бы «складывается», оставляя после себя лишь пунктирную линию с символом «**+**» в первой позиции. Полезность такого «складывания» становится очевидной после запуска, например, **'vimdiff today.dmp yesterday.dmp'**, где **today.dmp** и **yesterday.dmp** – мегабайтные дампы одной и той же БД, полученные сегодня и вчера соответственно. Результат будет представлен в виде соседствующих по вертикали окон, каждое из которых покажет только не совпадающие фрагменты файлов: совпадающие будут «сложены».

Не думаю, что список получился исчерпывающим, но для начала и этого более чем достаточно.

А вот ещё несколько менее оригинальных, но весьма полезных возможностей:

- поиск в произвольном направлении лексемы «под курсором»;
- поиск «определения» лексемы «под курсором»;
- полностью настраиваемая подсветка синтаксиса для множества типов файлов с возможностью создания собственных. В настоящее время в стандартную поставку входит свыше четырёхсот файлов-описателей синтаксиса языков программирования, конфигурационных файлов и журналов;
- всеобъемлющие help-файлы плюс руководство пользователя;
- встроенный язык сценариев для добавления новых возможностей;
- система plugin-ов;
- ввод специальных символов комбинацией двух символов, причём возможно определение собственных комбинаций;
- автоматическое определение типа файла (DOS, Mac, Unix) с возможностью сохранения в любом из этих форматов;
- запись действий пользователя в виде макросов – для выполнения повторяющихся операций;
- выделение символов, строк и прямоугольных блоков;
- редактирование в «виртуальном» режиме, когда пустые позиции автоматически заменяются пробелами – незаменимое средство при составлении таблиц;
- замена символов табуляции на пробелы;
- повторение последнего действия, каким бы сложным оно ни было;
- интеграция с Perl, Tcl и Python: можно расширить функциональность редактора, написав функцию на языке, более предпочтительном, чем собственный язык *VIM*, а можно просто выполнять какие-то действия якобы в любимом интерпретаторе, используя, на самом деле, модуль того же редактора. И многое, многое другое.

Программистам от программистов

Бытует мнение, что *VIM* написан «программистами для программистов». Не думаю, что это настолько справедливо, что не-программист не найдёт в *VIM* ничего полезного. Тем более, что сейчас уже совершенно неясно: увлеченно «ваяющего» личную web-страничку уже нужно относить к программистам или ещё нет? Тем не менее, не заметить, что именно программисты – наибольшие поклонники *VIM*, невозможно. В качестве иллюстрации ниже следует описание одного сеанса работы. Итак...

- Запуском vim без параметров я заставляю его предварительно считать **~/.viminfo**. В результате к моменту начала редактирования уже открыты три файла, с которыми я работал в последний раз. Убеждаюсь в этом, нажимая F12 и переключая при этом буфера: в **~/.vimrc** имеется строка

```
map! <F12> <Esc>:bn<CR>
```

которая для всех режимов (**!**) по нажатию F12 (**map <F12>**) выполняет (на всякий случай) перевод в нормальный режим (Esc) и ввод команды (**:bn** – вывести в текущее окно следующий в списке буфер). Ввод команды завершается **<CR>**.

- Поскольку опция **set viminfo** всё в том же **~/.vimrc** содержит, среди прочего, и параметр **f1** (сохранять файловую отметку **'0**, соответствующую текущему положению курсора в момент завершения последнего сеанса), то для всех буферов (файлов) курсор будет установлен в ту позицию, где он находился перед закрытием.
- Нажатием F5+S (в **~/.vimrc** имеется **'map! <F5> <Esc><C-W>'**, которая «привязывает» к нажатию F5 все команды манипуляций с окнами) получаю два расположенных друг под другом окна, в каждом из которых по одному нужному в настоящий момент буферу (файлу). Разумеется, размер окон меняется. Разумеется, ненужное окно всегда можно закрыть нажатием F10 (в **~/.vimrc** имеется 'map! **<F10> <Esc>:q<CR>'**). На буферах это, кстати, никак не отражается, и вплоть до закрытия последнего окна об их судьбе можно не беспокоиться. При

А ДАЛЬШЕ...

Вполне вероятно, что когда-нибудь вам потребуется пересобрать *VIM*. Меня, во всяком случае, когда-то не устроили сборки ни RedHat, ни Alt, ни ASP. Было это уже давненько, возможно, положение и изменилось, но я по-прежнему предпочитаю *VIM* собирать сам – всё-таки это самая используемая программа. Исходники находятся на **http://www.vim.org/**.

Современные версии *VIM* успешно «разговаривают» по русски, а вот за русским help и User Guide стоит сходить на **http://www.sf.net/projects/ruvim**. Эти переводы несколько устарели – вряд ли стоит заменять ими оригинальные файлы, но для чтения они незаменимы.

Вне конкуренции находится проект Vim Online (адрес: **http://vim.sourceforge.net/**). Возможно, именно с него стоит начинать, если вы никак не определитесь, нужен ли вам *VIM*. Свыше тысячи советов, почти полторы тысячи скриптов. И чего только с помощью *VIM* не делают...

Сайт Гомера Джила (Thomer M. Gil) – ещё одна общепризнанная точка старта в *VIM*-путешествие. Адрес: **http://thomer.com/vi/vi.html**.

Список русскоязычных ресурсов также достаточно внушителен. На **CitForum.ru** можно прочитать «*Поваренную книгу VIM*» (**http://citforum.ru/open_source/cook_vim**).

А на **www.vnc.org.ua** лежит перевод полновесного HOW-TO.

Почти хрестоматийный «*Путь к VIM*» Ильи Ялового можно найти по адресу: **http://www.opennet.ru/docs/RUS/go_to_vim/**.

Хороший справочник «*VIM – кратко обо всём*» предлагает Роман Савоченко: **http://diyaorg.dp.ua/oscadawiki/wakka.php?wakka=RomanSavochenko/VimShortAll**.

Автор также внёс свою лепту:

- «Vim. Может он действительно лучший?»
- «Vim-2 или «что может быть проще?»
- «Vim: уже шесть!»
- «Vim. По-прежнему: один из лучших»

закрытии последнего окна, которое логически соответствует выходу из программы, *VIM* напомнит о наличии в памяти буферов, сохранение которых в файл не выполнялось.

• Сохранение, как довольно частую операцию, я «привязал» к F2 (в **~/.vimrc** имеется **'map <F2> <Esc>:w<CR>'**).

• Разумеется, *VIM* «раскрашивает» содержимое файлов в соответствии с синтаксисом соответствующего языка. Практически невозможно сделать ошибку в написании оператора или базовой библиотечной функции, пропустить символ, отделяющий комментарий и т.п. В своё время, *VIM* «помирил» мою лень с рекомендацией давать переменным, функциям, классам и методам полные, исчерпывающие имена: почему бы и нет, если это делается один раз, а впоследствии автодополнение сделает ненужным полный набор, исключая при этом вероятные ошибки.

• Разумеется, *VIM* выполняет «autoindent» – блоки программы автоматически выделяются соответствующими отступами. Ввод закрывающей скобки любого типа буквально на секунду услужливо перенесёт курсор на соответствующую открывающую скобку, если она находится в пределах экрана. Если же конструкция достаточно сложна и соответствующую открывающую скобку «ещё поискать», то на помощь придёт «matching»: нажатие '%' в нормальном режиме и курсоре на скобке переносит курсор на соответствующую скобку.

• Потребуйся мне невзначай код какого-либо символа, нажатие g a даст полную информацию о символе под курсором.

• Операции с регулярными выражениями – на очень высоком уровне. Не Perl, но для редактирования – более чем достаточно. Поиск и замена становятся столь эффективны, что на приведение в «приличный» вид мегабайтного html-файла из-под MS Word (а вы когда-нибудь видели, как выглядит html в исполнении Word?) уходит 10-15 минут.

Программисты, как и пешеходы, составляют лучшую часть населения, однако, в отличие от последних, отнюдь не большую. Поэтому описание прелестей *VIM* для программистов на этом прекращаем и переходим к ответу на вопрос: насколько этот «дедушка»-редактор вписывается в современное графическое рабочее окружение?

А как в X-ах?

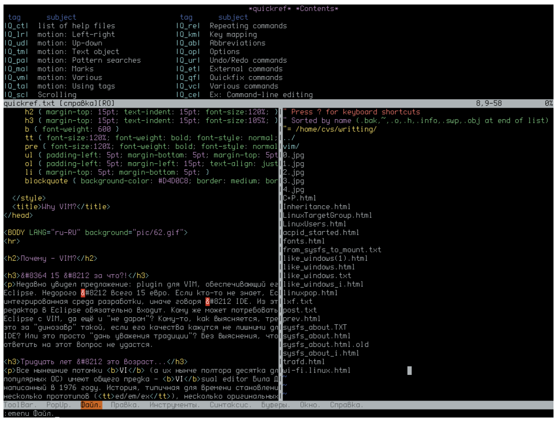
С воцарением в Linux framebuffer и всё большим распространением LCD мониторов стоит, возможно, пересмотреть отношение к редактированию текста в консольном режиме. В самом деле: поле редактирования – 128x50 символов при диагонали 15», цвет фона и символов – любой, частота кадров – 70 Гц, да и не важно это для LCD. Terminus – практически безукоризнен в качестве моноширинного фонта, а для текстов программ как раз моноширинный-то и предпочтителен... Вернёмся, однако, к X-ам.

Как это ни странно, но терминал-ориентированный *VIM* и в графической среде достаточно неплох, причём во всех доминирующих графических окружениях. Сам по себе *VIM*, скомпилированный с соответствующими опциями (а именно так большинство хранителей и делает) и получивший ещё одно (на сей раз символическое) имя – *gvim*, использует **Gtk**, но особого сродства к Gnome при этом не демонстрирует (разве что иконки указывают на это). Разработчики KDE благоволили к *VIM* настолько, что долгое время поддерживали проект под названием *KVim*, но это, к сожалению, в прошлом. В настоящее время в центре управления KDE (ветка «Компоненты KDE») можно найти позицию «Встраивание *VIM*», воспользовавшись которой, можно сделать *gvim* средством редактирования «по умолчанию», придав ему при этом некоторую Qt-шность. Два ряда иконок (одна от *gvim*, одна – от KDE) выглядят несколько курьёзно, но не будем привередливы: в любом случае, это всё тот же *VIM*. Более-менее существенные отличия сводятся к трём пунктам:

• поддержка мыши (в консольном варианте её обеспечивает *gpm*, соответственно, она опциональна);

• меню, которое в консольном варианте также опционально, для *gvim* – неотъемлемый компонент;

• использование возможностей *X Window* (доступ к общему буферу обмена без использования специальных регистров, произвольно задаваемые шрифты и т.п.).



Консольный VIM: начало начал.

Замечательно, что помимо **~/.gvimrc** (а как, вы полагали, конфигурируется *gvim*?), учитываются все опции, перечисленные в **~/.vimrc**. Таким образом, опыт и приёмы консоли практически наследуются. А вдобавок мы получаем набор иконок для самых распространённых действий и меню, настолько исчерпывающее, насколько вы сами или хранитель дистрибутива позаботились об этом.

Особенно радует ожившее колёсико мыши (*gpm* его не поддерживает, а вот *X* – вполне) и позиционирование курсора щелчком левой кнопкой мыши. При этом привычные способы выделения (начало блока – клик левой кнопкой, конец – правой) и вставки (нажатием средней кнопки-колесика) работают по-прежнему.

Настоящим продолжателем традиций *VIM* в графической среде может стать *YZIS*, а, точнее, *kyzis* – его *Qt*-реализация, доступная в Сети по адресу **http://www.yzis.org/**. Проект пока далёк от завершения, но уже сейчас можно видеть, что он удачно сочетает в себе графическую лаконичность Qt и идеологию *VIM*.

Одним словом, в графической среде *VIM* ничего не потерял и даже кое-что приобрёл. Остаётся только пожелать того же другим Unix-»дедушкам».

Стоит ли беспокоиться?

Убеждать пользователя в необходимости изучения того или иного программного продукта – занятие бесперспективное. Тот, кем движет любопытство, в подобном не нуждается, а «всем довольному» – какой смысл затрачивать дополнительные усилия? Освоение сколько-нибудь сложного инструмента становится необходимостью только тогда, когда обещает выигрыш в работе. Множество программистов во всём мире считают использование *VIM* в своей работе оправданным. Каждый из них когда-то решил, что затраты на изучение хорошего инструмента пренебрежимо малы в сравнении с ожидаемой пользой. А каждый новый потенциальный пользователь будет решать этот вопрос для себя сам.

На первый взгляд, знакомство лучше начинать с *gvim* или *kyzis* – из них хоть выйти можно без поиска соответствующей команды. Но, в то же время, далеко не очевидно, чем такой «причёсанный» *VIM* лучше *joe* или *mcedit*. Так что сначала лучше бы почитать что-то вроде этой статьи: где-нибудь да отложится информация о существовании у *VIM* неких неординарных возможностей. Далее стоит поискать задачу, для решения которой в *joe* или *mcedit* нужно так долго нажимать клавиши, что и приступать к этому не хочется. Ну, а потом можно попытаться эту задачу решить с помощью *VIM*.

И если задача будет-таки решена и инструмент продемонстрирует, что стоил потраченного на знакомство с ним времени, – тогда имеет смысл перейти к его «приручению». Лучшие, на мой взгляд, источники информации для этого перечислены во врезке «А дальше...»

Недостатка в документации нет. И хотя ни один из перечисленных источников в полной мере не отражает оригинальную документацию на английском, расстраиваться по этому поводу не стоит – в полной объёме она вряд ли кому-то и нужна. *VIM* слишком универсален, чтобы одному человеку потребовалось «сразу всё». LXF

ПРОГРАММИРОВАНИЕ СЦЕНАРИЕВ

Уроки Python

ЧАСТЬ 4 Мы вплотную подошли к наиболее интересной теме – реализации объектно-ориентированного подхода в языке Python. Репортаж с места событий ведет **Сергей Супрунов**.

МЕСЯЦ НАЗАД

Мы научились писать функции и модули.

Немного терминологии

ООП (объектно-ориентированное программирование), наряду с анализом и дизайном – это важнейшая часть очень популярного объектного подхода к разработке программного обеспечения. В данном случае в дизайне программы выделяются так называемые объекты, которые обладают атрибутами (свойствами) и методами (способами изменения свойств).

Новые объекты создаются на базе классов, которые являются своего рода шаблонами, описывающими общие характеристики объектов (также называемых экземплярами класса).

Для ООП характерны следующие принципы:

- инкапсуляция: экземпляр класса рассматривается как «чёрный ящик», когда его внутреннее устройство скрыто, а всё взаимодействие с объектом выполняется через предусмотренный для этого интерфейс, т.е. набор методов. Нужно заметить, что в *Python* программист, в принципе, имеет возможность непосредственно воздействовать на свойства объекта, минуя интерфейс – здесь инкапсуляция реализована не на уровне ограничений языка, а на уровне соглашений.
- наследование: вы можете создавать новые классы на базе существующих, при этом они будут получать все атрибуты и методы, описанные для родительских классов. Наследование – это наиболее естественный путь разработки классов, подобных родительским, но с дополнительным набором свойств.

Есть и другие принципы (полиморфизм, агрегация и т.д.), с которыми вы сможете глубже познакомиться в специальной литературе.

Реализация ООП в Python

В отличие от Perl, в *Python* объектный подход заложен в основу этого языка. Здесь почти всё является объектами, даже строки и числа. Не верите? Смотрите сами:

```
>>> print 1.0.__add__(5)
6.0
```

То есть мы применили к числу 1.0 (объекту) метод **__add__()**, который является «внутренней» реализацией операции сложения. Нужно заметить, что здесь из-за ограничений синтаксиса мы вынуждены использовать число с плавающей запятой, поскольку первая точка воспринимается интерпретатором именно как разделитель целой и дробной частей, а вот вторая уже отделяет объект от имени метода. Уже известная нам функция **dir(3)** вернёт ещё 52 метода, которые вы можете применять к числу.

Для определения класса используется специальный оператор class:

```
#!/usr/bin/python
# file: ctest.py
class User:
  comment = 'User'
  def __init__(self, username):
    self.username = username
    self.password = ''
  def setpass(self, value=''):
    self.password = value
  def checkpass(self, typed):
    if typed == self.password:
      return 1
    else:
      return 0
```

Здесь мы определили класс, экземпляры которого будут хранить информацию о пользователях (имя и пароль). Специальный метод **__init__()**, играющий роль конструктора, автоматически исполняется при создании нового объекта. В нём мы присваиваем атрибуту username значение, переданное конструктору как параметр, и создаём атрибут password с пустым значением.

Обратите внимание на обязательность использования параметра **self**, который должен быть первым в описании любого метода. При работе с конкретным объектом здесь будет указываться его идентификатор, чтобы интерпретатор мог определить, с каким же объектом он имеет дело.

Описанные далее методы **setpass()** и **checkpass()** служат для того, чтобы установить значение атрибута password и сравнить с ним значение, введённое пользователем. На практике работа с нашим классом может выглядеть таким образом:

```
u1 = User('Vasya')
u1.setpass('qwerty')
userpass = raw_input('Enter your password:')
if u1.checkpass(userpass):
  print 'Password is OK'
else:
  print 'Password is wrong'
u1.password = 'sasdf'
```

В первой строке мы создали объект (экземпляр класса User). Затем мы задали ему пароль и чуть позже сравнили его с тем, который пользователь ввёл по запросу сценария.

Последняя строка демонстрирует, что пароль можно изменить и непосредственно. Это является нарушением принципа инкапсуляции. Язык *Python* предусматривает два соглашения для решения этой проблемы: во-первых, если имя атрибута или метода начинается с символа подчёркивания, это является признаком того, что они предназначены для «внутреннего потребления» и не должны использоваться непосредственно. Однако сам *Python* никак не ограничивает это. Если вы желаете получить более жёсткий контроль, используйте имена, начинающиеся двумя символами подчёркивания. Такое имя уже не будет доступно через пространство имён объекта (хотя обходной путь всё же есть).

Особое место занимают специальные методы (такие как показанный выше метод **__init__()**). Эти методы реализуют ряд «сервисных» функций (например, конструктор **__init__()**, деструктор **__del__()**, отвечающий за корректное удаление объекта, и т.п.), а также позволяют переопределить поведение, заложенное в интерпретаторе. Например, метод **__add__()** исполняется при обработке оператора сложения «+». Определим его для некоторого класса:

```
#!/usr/bin/python
# file: stest.py
class Test:
  def __init__(self, value):
    self.value = value
  def __add__(self, other):
    return self.value – other
```

И теперь, вместо ожидаемого сложения, мы увидим вычитание:

```
>>> from stest import Test
>>> a = Test(5)
>>> print a + 3
2
```

Естественно, этим не стоит злоупотреблять, но вы должны знать, что очень многие «стереотипы» становятся весьма условными, если речь заходит о *Python*.

Нужно заметить, что вы можете задавать атрибуты объекта динамически, что называется, «на лету». Продолжим предыдущий пример:

```
>>> a.description = 'Описание'
>>> print a.description
Описание
```

Хотя атрибут **description** отсутствует в классе Test, мы можем работать с ним, как с обычным. Таким образом можно задать и атрибуты, имена которых начинаются с двух подчёркиваний. Это создаёт видимость того, что никаких ограничений на эти имена нет, но на самом деле получится совершенно иной атрибут, нежели определённый в описании класса. Так что не увлекайтесь подчёркиваниями.

Наследование

Несколько слов нужно сказать о наследовании. Рассмотрим пример:

```
#!/usr/bin/python
# file: ntest.py
from ctest import User
class ShellUser(User):
  def __init__(self, username):
    User.__init__(self, username)
    self.shell = '/bin/sh'
  def setshell(self, newshell):
    self.shell = newshell
u2 = ShellUser('Petya')
u2.setpass('12345')
print u2.password, u2.shell
```

Как видите, дочерний класс ShellUser получает все свойства родительского (имена родительских классов перечисляются в скобках в определении class). Теперь мы можем расширить этот класс новыми атрибутами и методами, и в дальнейшем использовать их наряду с определёнными в родительских классах.

Обратите внимание, что при создании экземпляра нового класса исполняется только его метод **__init__()**, инициализацию родительского класса нужно вызывать явно.

Классы в стандартных модулях

Классы очень широко используются в стандартных модулях *Python*. В качестве примера рассмотрим один достаточно полезный модуль – **StringIO**. С его помощью вы можете применять к строкам методы работы с файлами (**read()**, **write()** и т.д.). Это может быть необходимо в тех случаях, когда другой метод или функция может обрабатывать только файлы. Например, класс Message модуля **rfc822**, с помощью которого, в частности, можно разбирать сообщения электронной почты, умеет работать только с файлом, содержащим текст сообщения.

Рассмотрим пример:

```
#!/usr/bin/python
import rfc822, StringIO
mailstr = """\
From: user@domain.ru
To: me@mysite.ru
Subject: Test Message
Hello!
It is a test message.
"""
fileobj = StringIO.StringIO(mailstr)
message = rfc822.Message(fileobj)
print "%s wrote:\n" % message.getheader('From')
bodyfrom = message.startofbody
fileobj.seek(bodyfrom)
body = fileobj.read()
print body
```

Итак, что здесь происходит? Переменная mailstr содержит текст в формате почтового сообщения. Поскольку модуль **rfc822** умеет работать только с файловыми объектами, мы создаём экземпляр класса **StringIO** под именем **fileobj**. На его основе уже создаётся объект message класса Message, описанного в модуле **rfc822**.

С помощью метода **getheader()** мы распечатываем значение поля «From», а узнав из атрибута **startofbody** объекта message, где начинается тело сообщения (отделённое от заголовка пустой строкой), мы видим, что нашему объекту **fileobj** не чужды никакие методы настоящих файлов – можно позиционировать указатель (метод **seek()**), считывать содержимое от текущей позиции до конца файла (**read()**), и т.д.

ЧЕРЕЗ МЕСЯЦ

Мы подробно рассмотрим обработку текста – методы строк, регулярные выражения, форматирование вывода и т.п.

Документируй это!

Хотя в языке *Python* инкапсуляция, т.е. сокрытие внутреннего устройства класса, не реализована в чистом виде (вы можете обращаться напрямую к любому атрибуту, за небольшим исключением), но всё же хорошим тоном считается работать с объектами классов только через предоставляемый ими интерфейс. И в этих условиях особое значение приобретает документация. В *Python* реализован очень удобный способ, скажем так, «онлайновой» документации – первая текстовая строка класса (или функции) воспринимается как его описание, которое может быть в любое время получено с помощью атрибута **__doc__**:

```
>>> class Cla:
...   "Документация класса"
...   pass
...
>>> print Cla.__doc__
Документация класса
```

Для многострочных описаний удобно использовать утроенные кавычки.

На этом мы завершим наше знакомство с классами *Python*. Естественно, на таких небольших примерах их мощь и удобство остались за кадром, хотя при работе со стандартными модулями в любом случае необходимо понимать, как всё это работает. Настоящий же эффект от использования объектно-ориентированного подхода вы ощутите при работе над большими проектами. **LXF**

СЕРИЯ «КОМПЬЮТЕРНЫЕ TeXнологии»

MetaPost Начала автоматизации

ЧАСТЬ 3 Компьютер не умеет читать ваши мысли, зато неукоснительно следует инструкциям. **Евгений Балдин** научит вас отдавать правильные команды и извлекать из этого выгоду.

МЕСЯЦ НАЗАД

Мы познакомились с базовыми элементами Metapost.

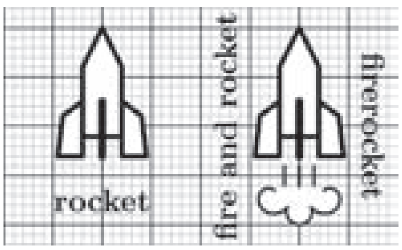
До сего момента мы концентрировались на том, как объяснить компьютеру, чтобы он сделал то или иное движение. Теперь воспользуемся способностью компьютера помнить предыдущие действия и извлекать их из памяти по мере необходимости. Автоматизация рутинных процедур это то, для чего компьютеры и предназначены. Практиковаться в автоматизации следует постоянно. Несмотря на затраченное на обучение время, в результате время же и экономится.

Объекты picture

В процесс повествования объект picture или картинка уже упоминался. Картинка представляет из себя совокупность путей и точек, которую можно подвергать трансформации. В уже существующие картинки можно добавлять пути, замкнутые области и другие картинки.

Для начала опять же воспользуемся миллиметровкой для отрисовки какого-либо рисунка, например, ракеты:

Ракета может быть без выхлопа (rocket) и с выхлопом (firerocket).



В процессе создания firerocket был использован рисунок самого выхлопа (fire).

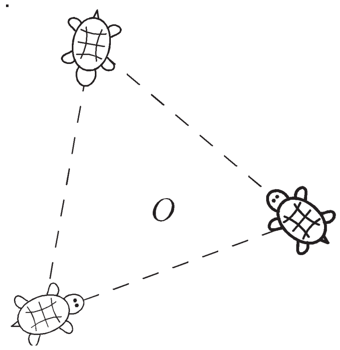
```
%Файл picture.1.mp
%Ракета без выхлопа 10x12 Центр у стабилизаторов
picture rocket;
rocket:=nullpicture;
addto rocket contour (-2,-1)--(-2,6)--(0,10)--(2,6)--
 (2,-1)--cycle withpen pencircle scaled 0.4 withcolor white;
addto rocket doublepath (-2,-1)--(-2,6)--(0,10)--(2,6)--
 (2,-1)--cycle withpen pencircle scaled 0.5;%Корпус
addto rocket contour (-2,2.5)--(-4,1)--(-4.5,-3)--
 (-2,-3)--cycle withpen pencircle scaled 0.4 withcolor white;
addto rocket doublepath (-2,2.5)--(-4,1)--(-4.5,-3)--
 (-2,-3)--cycle withpen pencircle scaled 0.5;%левая дюза
addto rocket contour (2,2.5)--(4,1)--(4.5,-3)--(2,-3)--cycle
 withpen pencircle scaled 0.4 withcolor white;
addto rocket doublepath (2,2.5)--(4,1)--(4.5,-3)--
 (2,-3)--cycle withpen pencircle scaled 0.5;%правая дюза
addto rocket doublepath (0,2.5)--(0,-3)
 withpen pencircle scaled 0.8;%центральная дюза

%выхлоп
picture fire;
fire:=nullpicture;
addto fire doublepath (0,-4)--(0,-6)
 withpen pencircle scaled 0.3;%выхлоп 1
addto fire doublepath (-1.5,-4)--(-1.5,-6)
 withpen pencircle scaled 0.3;%выхлоп 2
addto fire doublepath (1.5,-4)--(1.5,-6)
 withpen pencircle scaled 0.3;%выхлоп 3
addto fire contour (-2.5,-6.5){dir 135}..(-4,-8)..
 {dir 50}(-1.2,-8.2){dir -110}..(0,-10)..
 {dir 110}(1.2,-8.2){dir -50}..(4,-8)..
 {dir -135}(2.5,-6.5)--cycle withpen pencircle scaled 0.4
 withcolor white;
addto fire doublepath (-2.5,-6.5){dir 135}..(-4,-8)..
 {dir 50}(-1.2,-8.2){dir -110}..(0,-10)..
 {dir 110}(1.2,-8.2){dir -50}..(4,-8)..{dir -135}(2.5,-6.5)
 withpen pencircle scaled 0.3;%облако

%ракета и выхлоп
picture firerocket;
firerocket:=rocket;
addto firerocket also fire;
```

Прежде чем что-то добавить к картинке, её необходимо инициализировать. В MetaPost есть две определённые по умолчанию картинки: nullpicture — пустая картинка и currentpicture — текущая картинка. Пользуясь последней переменной, можно в любой момент сохранить результаты промежуточной отрисовки. Добавление элементов к картинке производится с помощью инструкции addto, после которой указывается картинка, к которой и добавляется тот или иной элемент. Путь добавляется с помощью инструкции doublepath, замкнутая область - с помощью инструкции contour, а другая картинка с помощью инструкции also.

Ранее был создан рисунок черепашки. Для его обозначения была выбрана переменная **Turtle**. Теперь с ней можно поработать, как с единым элементом, например, для иллюстрации задачи: «Черепашки расположены в углах правильного треугольника со стороной a и всегда ползут в направлении своей соседки против часовой стрелки со скоростью v. Когда они встретятся?»



Картинку можно отобразить с помощью команды **draw**. Над картинкой можно производить различные преобразования. В данном случае картинка поворачивалась, масштабировалась и сдвигалась.

```
%Файл pic.mp
beginfig(17) ;
 numeric u;u = 0.8mm;
 numeric dphi; dphi=20;
 draw 30u*dir (90+dphi)--30u*dir
(210+dphi)--
   30u*dir (330+dphi)--cycle dashed evenly scaled 1u;
 draw Turtle rotated (-120+dphi) scaled 1u
            shifted (30u*dir (90+dphi));
 draw Turtle rotated dphi scaled 1u
            shifted (30u*dir (210+dphi));
 draw Turtle rotated (120+dphi) scaled 1u
            shifted (30u*dir (330+dphi));
endfig ;
```

Обратите внимание, что линия, соединяющая черепах, нарисована пунктиром. Определённая по умолчанию переменная evenly тоже является картинкой, поэтому её можно масштабировать с помощью декларации scaled. То есть, если вам нужен более широкий шаг пунктира, то вместо масштаба 1u можно указать 2u. Если вас не устраивает где располагаются штрихи у штриховки, то можно воспользоваться декларацией сдвига shifted.

Кроме шаблона evenly в MetaPost определён шаблон withdots, который позволяет рисовать кривую с помощью точек.

Вы можете определить свой шаблон для пунктира примерно следующим образом:

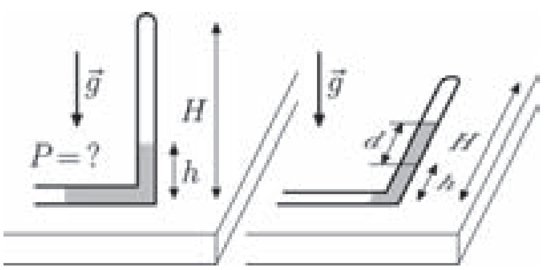
```
picture dash_center;
dash_center:=dashpattern(on 3 off 1.5 on 0.5 off 1.5);
draw 30u*dir (90+dphi)--30u*dir (210+dphi)--
  30u*dir (330+dphi)--cycle dashed dash_center scaled 1u;
```

Функция dashpattern принимает список on/off с числовой информацией в какой момент рисовать/не рисовать. В этом примере определён шаблон для штрих-пунктирной линии, которая обычно используется для обозначения оси симметрии.

Трансформация

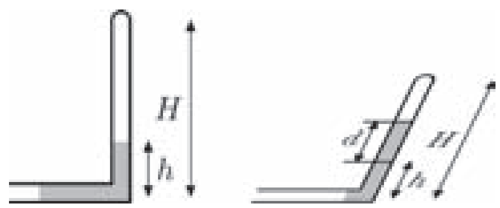
К задаче N 3 варианта ГГФ-51в требовалось изобразить L-образную трубку с водой. По условию, трубка сначала стояла вертикально, а потом была положена на стол.

Чтобы схематично это изобразить, вовсе необязательно уметь работать в трёхмерном редакторе. Ниже идёт код, который рисует вертикально стоящую пробирку с размерами, а затем наклоняет её.




```
%Файл transform.mp
%пример использования slanted
beginfig(1) ;
 numeric u;
 u = 0.8mm;
 %пробирка
 cutdraw (0u,0u)--(20u,0u)--(20u,30u){dir 90}..
   {dir -90}(17u,30u)--(17u,3u)--(0u,3u)
  withpen pencircle scaled 0.5u;
 drawdblarrow (23u,10u)--(23u,1u);
 label.rt(btex \(h\) etex,1/2[(23u,10u),(23u,1u)]);
 drawdblarrow (30u,30u)--(30u,1u);
 label.lft(btex \(H\) etex,1/2[(30u,30u),(30u,1u)]);
 picture Base;
 Base:=currentpicture; %запоминаем
 clearit; %очищаем текущую картинку
 %рисуем воду когда пробирка будет наклонена
 fill (15u,0u)--(20u,0u)--(20u,20u)--(17u,20u)--
  (17u,3u)--(15u,3u)--cycle withcolor 0.7white;
 draw Base;
 draw (12u,20u)--(20u,20u);draw (12u,10u)--(20u,10u);
 drawdblarrow (14u,20u)--(14u,10u);
 label.lft(btex \(d\) etex,(14u,16u));
 picture Slant;
 Slant=currentpicture; %запоминаем
 clearit; %очищаем текущую картинку
 %рисуем воду когда пробирка стоит
 fill (5u,0u)--(20u,0u)--(20u,10u)--(17u,10u)--
  (17u,3u)--(5u,3u)--cycle withcolor 0.7white;
 %отрисовываем пробирку
 draw Base;
 %отрисовываем пробирку и наклоняем её
 draw Slant yscaled 2/3 slanted 1/2 shifted (40u,0u);
endfig ;
```

В примере применяется возможность сохранить текущее состояние с помощью currentpicture, а так же возможность полностью очистить текущую картинку с помощью инструкции clearit.



Наклон вертикально стоящей пробирки происходит с помощью масштабирования yscaled и, собственно, наклона slanted.

MetaPost поддерживает следующие базовые линейные преобразования:

Команда	Результат
(x,y) shifted (a,b)	(x+a,y+a)
(x,y) scaled s	(sx,sy)
(x,y) xscaled s	(sx,y)
(x,y) yscaled s	(x,sy)
(x,y) slanted s	(x+sy,y)
(x,y) rotated θ	(x cosθ - y sinθ, x sinθ + y cosθ)
(x,y) zscaled (a,b)	(xa-yb, xb+ya)

Кроме перечисленных базовых преобразований полезными для использования являются макросы rotatedaround ((a,b), θ) — поворот вокруг точки (a,b) на угол θ и reflectedabout (z1,z2) — отражение относительно линии, проходящей через точки z1 и z2.

MetaPost поддерживает объекты типа transform, то есть можно определить любое необходимое для вас преобразование, чтобы использовать его в дальнейшем.

```
transform t;
t:= identity yscaled 2/3 slanted 1/2 shifted (40u,0u)
draw Slant transformed t;
```

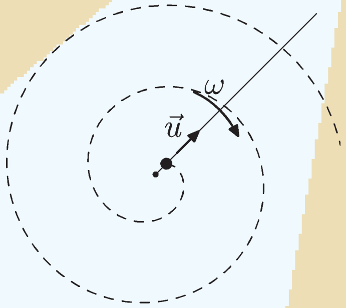
Используемая при описании преобразования t константа identity тоже является преобразованием. identity – это «пустое» преобразование, то есть преобразование, которое ничего не делает (математически, оператор такого преобразования описывается единичной матрицей – identity matrix, что и определяет название).

Циклы и условные операторы

Циклы и условные операторы в MetaPost отличаются от того, что обычно есть в других языках программирования. Цикл не просто повторяет перечисленные в теле цикла инструкции — он дублирует текст, то есть внутри цикла не обязательно должна находиться синтаксически законченная конструкция. Это же относится и к условным операторам.

«Шарик с постоянной скоростью движется вдоль спицы, которая с вращается с постоянной угловой скоростью. Требуется изобразить траекторию шарика.»

Для изображения траектории надо построить минимодель явления и задать физические параметры: поступательную скорость вдоль спицы v, угловую частоту w и начальные условия r и phi. Сама траектория создаётся с помощью следующего кода:




```
%Файл cycle.mp
v:=27u;w:=360;N:=2.1;phi:=45;r:=5u;n:=100;
path p; pair O;
O:=(r*cosd(phi),r*sind(phi));
p:=O for i=0 upto n:
      ..((r+v*N*i/n)*dir(-w*N*i/n)+phi))
   endfor;
draw p withpen pencircle scaled 0.5u
                        dashed evenly scaled 1u;
```

Декларация upto - это сокращение для step 1 until. Аналогично downto является сокращением для step -1 untull.

Формальный синтаксис цикла представлен ниже:

```
for i=x1 step x_2 until x3: text(i) endfor
```

Это одна из форм, которая поддерживается META. Ещё одна форма представляет бесконечный цикл:

```
forever: "текст" endfor
```

Для того чтобы выйти из подобного цикла необходимо воспользоваться конструкцией вида:

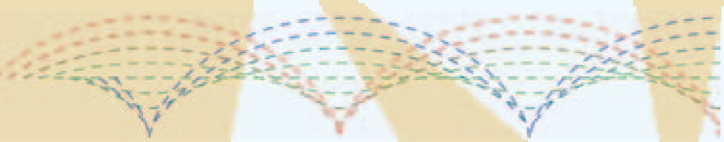
```
exitif ("булево выражение")
```

Булево выражение может быть переменной типа boolean (true/false) или результатом сравнения чисел, точек, путей или преобразований. Операторы сравнения почти совпадают с операторами сравнения языка C, за исключением оператора равенства «=» и оператора неравенства «<>». Выражение можно инвертировать с помощью приставки not и объединить с другим с помощью приставок and или or.

Формальный синтаксис условного оператора представлен ниже:

```
if ("булево выражение1"): "текст1"
elseif ("булево выражение2"): "текст2"
else: "текст3" fi
```

Воспользуемся циклами для изображения циклоиды — траектории точки на катящемся колесе.



Обратите внимание, что в конце цикла или условного оператора нет необходимости ставить «;» это позволяет использовать их довольно изощрённым образом.

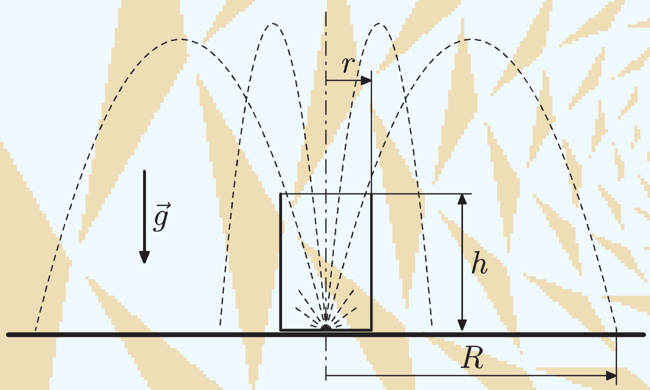
```
%Файл cycle.mp
%Рис к задаче 1.5.8 (20x120) - циклоида
beginfig(1) ;
  numeric u;u = 0.8mm;
  numeric R; R=10u;
  path p,cycl;
  p:=(-R,0u)..(R,0u)..cycle;
  %колесо
  draw p withpen pencircle scaled 0.3u
                        dashed withdots scaled 0.5u;
  numeric j,n,v,w,phi,nsteps;
  j=0;n=100;v=109.8u;w=-(v/R)*180/3.14;phi=180;nsteps=4;
  numeric r,i;
  for i:=0 upto 2*nsteps:
    r:=R-1/nsteps*R*i;
    %метки
    for j:=0 step n/4 until n:
      draw (j*(v/n)+r*cosd(j*(w/n)+phi),r*sind(j*(w/n)+phi))
        withpen pencircle scaled 1u;
    endfor;
    %траектория меток
    cycl:=for j:=0 upto n:
      if j<>0:..fi
        (j*(v/n)+r*cosd(j*(w/n)+phi),r*sind(j*(w/n)+phi))
    endfor;
    draw cycl dashed evenly scaled 1/2u
      withcolor (max(1-i/nsteps,0)*red+
           min(i/nsteps,2-i/nsteps)*green+
           max(i/nsteps-1,0)*blue);
  endfor;
endfig ;
```

В этом коде выражение if j<>0:..fi использовалось для того, чтобы перед первой точкой пути, описывающем циклоиду, не было декларации соединения. Я не знаю, какой еще из «популярных» на сегодня языков обладает такой способностью.

Макросы

Пользовательские функции в META фактически заменяются макросами. Как следствие, функции могут вернуть любую конструкцию: от числа до картинки.

Одним из моих ранних рисунков на META был «взрыв» в стакане. Требовалось изобразить траекторию «осколков» которые летят по параболе и найти самую дальнюю точку, которую достигают осколки при таком «взрыве».



Была написана процедура, которая рисовала параболу по переданным параметрам. Вызов выглядел примерно следующим образом:

```
Parabola_dashed(0u,0u,-1.25angle(20/sqrt(8),
          10*sqrt(8)),10*sqrt(8)*u,0,100,1);
```

Сам макрос для отрисовки параболы представлен ниже.

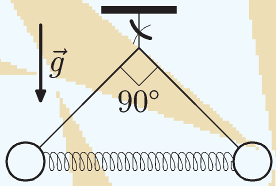
```
%Файл macros.mp
%Рисует параболу из точки (x,y) (полёт камня) штриховая
%линия. В качестве параметров передаётся (x,y), ang-угол,
%vel-скорость (100), %from,to - откуда и до куда рисовать
%параболу в процентах [0,100], mag - увеличение (0.8u)
%Для простоты g=10
def Parabola_dashed(expr x,y,ang,vel,from,to,mag) =
  path p;
  numeric t,g,n;
  picture dash_one;
  dash_one:=dashpattern(on 2mag off 2mag);
  n=100;%число шагов
  g=10.;
  t:=(2*vel*sind(ang)*from)/(g*n);
  p:=(vel*cosd(ang)*t*mag,(vel*sind(ang)*t-g*t*t/2)*mag);
  for i=from+1 upto to:
    t:=(2*vel*sind(ang)*i)/(g*n);
```

```
    p:=p..(vel*cosd(ang)*t*mag,
        (vel*sind(ang)*t-g*t*t/2)*mag);
  endfor;
  draw p  shifted (x*mag,y*mag) dashed dash_one;
enddef;
```

Не самое удачное решение, но оно выполняло то, что от него требовалось. В подобных случаях лучше, чтобы в результате деятельности макроса оставался объект, который потом можно нарисовать с помощью команды draw и трансформировать по мере необходимости. Тогда функции передавалось бы гораздо меньше параметров, что значительно бы все упростило. Например, вызов для рисования пружинки, которая необходима в следующем примере, использует всего три входных параметра:

```
draw Spring(25u,1.2u,25) rotated -90
  shifted (-12.5u,-20u) withpen pencircle scaled 0.1u;
```

«Два тела, соединённые пружинкой, висят в поле тяжести на нитях, образующих угол в 90⁰. В какой-то момент нить, крепящую конструкцию к потолку, разрывают.»



Надо нарисовать пружинку. Изображение пружинка может пригодится много где ещё, поэтому оно было оформлено как макрос.

```
%Файл macros.mp
%Создаёт пружину, высоты h, радиуса r, с числом витков n
% (0,0) - в основании пружины
vardef Spring(expr h,r,n) =
  begingroup save i;
    (0,0)--(0,-r/2+0.5h/n){dir 180}
    for i=h/n  step h/n until h:
      ..tension 1.2..(-r,i-h/n)..tension 1.2 ..
      (0,r/2+i-0.5h/n)..tension 1.2 ..(r,i)..
      tension 1.2 ..(0,-r/2+i+0.3h/n){dir 180}
    endfor--(0,h)
  endgroup
enddef;
```

Обратите внимания на инструкцию tension — натяжение. Она говорит с какой «силой» надо «натянуть» соединение между точками. Значение 1.2 означает, что это следует сделать чуть потуже, чем обычно. С помощью этой инструкции описывается соединение между точками в определении пути типа «натянутая прямая»:

```
  def --- = ..tenstion infinity.. enddef;
```

Если отрисовка параболы является аналогом процедуры, то создание пружины аналогом функции. Вызовы begingroup endgroup позволяют обособить вычисления, проводящиеся между ними, от «внешнего мира». С помощью команды save можно защитить переменные внутри группы — «сохранённые» таким образом переменные восстанавливают свои значения после выхода за пределы endgroup.

Параметры, которые передаются внутрь макроса, перечисляются после декларации expr. Чтобы что-то вернуть в результате исполнения макроса, возвращаемое выражение надо поместить в конце макроса без завершающего символа «;».

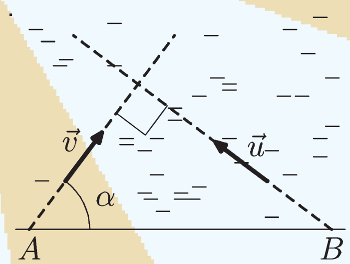
Отличие vardef от def заключается в том, что в случае def в качестве названия макроса передаётся «символьная лексема», а в случае vardef «объявляемая переменная». Отличие между этими понятиями заключается в том, что объявляемая переменная может состоять из нескольких символьных лексем. Таким образом вы можете создавать переменные с модифицирующимися именами. Если вам этого не надо, то используйте def.

Средства поддержки макросов в MetaPost исключительно мощные и разнообразные. В частности, с помощью инструкции primarydef можно доопределить недостающие бинарные операторы.

Стандартные функции

Лучший способ облегчить себе жизнь при написании программы - это не писать её, а воспользоваться уже готовыми компонентами. META является специализированным языком, поэтому число стандартных функций не очень велико, но их выбор весьма показателен.

«Из точек A и B в море вышли два корабля...» Требуется изобразить поверхность воды:



При кодировании этого рисунка использовалась функция генерации случайных чисел:

```
uniformdeviate n
```

В результате выполнения функции получалось случайное число в интервале [0,1]. Кроме упомянутой функции в META есть ещё один генератор случайных чисел normaldeviate — он создает числа в соответствии с распределением Гаусса (~exp(–x/2)).

Хотелось бы упомянуть о возможности разлагать сложные объекты на составляющие, например:

```
numeric x[ ],y[ ];
pair A,B; A=(x1,y1);A=(x2,y2);
%A=(xpart x1,ypart y1)
color c; c=(r,g,b);
%c=(redpart c,greenpart c,bluepart c)
path p;
p=A--B;
%A = point 0 of p = point 2 of p
%B = point 1 of p = point length p of p
```

Таким образом можно «разобрать» на части любой путь, причём номер точки не обязательно должен быть целым (берётся точка на линии соединения в соответствии с дробной частью). С помощью функции length можно узнать число заданных точек в пути, а с помощью arclength — его длину.

К уже известным вычислительным функциям sqrt, abs, mod, round, sind и cosd полезно добавить mlog (f(x)=256 ln x) и mexp (f(x)=exp(x/256)).

Для операций с точками будут полезны функции angle (x,y) — вычисления угла наклона к оси абсцисс для вектора ((0,0)--(x,y)) в градусах (операция, обратная dir α) и unitvector (x,y) — единичный вектор из начала координат.

Полный список стандартных функций представлен в «A User's Manual for MetaPost» Джона Хобби. Этот текст идёт со стандартной поставкой LaTeX в виде файла **mpman.pdf**.

ЧЕРЕЗ МЕСЯЦ

Мы узнаем, как создавать красивые диаграммы и представительные графики.

CYRILLIC HOWTO (russian)
ИСТОРИЯ СВОБОДНОГО ПРОЕКТА

Свободные исходники – это прежде всего тексты.
Автор русского Cyrillic HOWTO **Евгений Балдин** рассказывает историю своего проекта

8 лет назад я попытался понять, что же я могу сделать для сообщества. Это должно было быть то, что пригодилось бы и мне. Так был начат проект Cyrillic HOWTO (russian). Чуть больше двух лет назад Cyrillic HOWTO (russian) перестал изменяться. За пять лет развития проекта большинство проблем, вызвавших его к жизни, канули в лету.

Хотелось бы поделиться опытом, предостеречь от возможных ошибок, да и просто рассказать историю. Это был свободный проект. Он существовал, пока был полезен. Его можно возродить, если на то возникнет необходимость.

Предыстория

С Linux я познакомился в конце 1997 года. Как раз появилась новая версия Slackware 3.4, которая и встала на мой домашний компьютер. Зачем мне это понадобилось, я уже и не вспомню. В любом случае полученные навыки не пропали даром. На сегодня Linux – это моя рабочая среда.

Одним из неприятных моментов было то, что при установке по умолчанию Linux того времени с русским языком «не дружил» совершенно. Информация о том, какие «приседания» необходимо сделать чтобы решить эту проблему, была разрозненной. Основным сборником этих сакральных знаний был «The Linux Cyrillic HOWTO» от Александра Беликова.

Этот замечательный текст входил в набор стандартных HOWTO и, как следствие, был написан на английском языке. Объяснение подобного выбора было простым: при установке Linux у вас есть только латинские буквы, поэтому HOWTO может быть только на английском, а человек, который не знает английского в совершенстве – не достоин работать с Linux. Кстати, это мнение существует и по сей день.

К счастью технический английский не так страшен, как представляется с первого раза и школьных знаний вполне хватает. А если не хватает, то перевод чего-нибудь на русский позволяет эти знания существенно пополнить. Первой своей жертвой в целях обучения я выбрал именно «The Linux Cyrillic HOWTO». Если вы хотите что-то сделать для сообщества, то делать это надо с пользой для себя.

История с комментариями

Декабрь 28, 1999 The Linux Cyrillic HOWTO Alexander L.Belikoff (Александр Л. Беликов), (abel@bfr.co.il), Bloomberg L.P.; Переводчик: Балдин Евгений Михайлович, (E.M.Baldin@inp.nsk.su), Новосибирск, Россия
Версия 4.2 b2 с добавлениями.

Вплоть до середины 2000 проект представлял из себя русский перевод оригинального документа с дополнениями. Даже в 1997 году оригинал был не во всём актуален. Но случилось то, что случается со многими проектами одного человека. Александр Беликов потерял интерес к своему «детищу». Это произошло не сразу. После нескольких попыток прояснить судьбу текста я решился перестать быть просто переводчиком и начал свою нумерацию версий.

Автор редко когда отказывается от поддержки проекта – ему кажется, что завтра, когда выдастся минутка, он всё доделает. Только эта минутка не появляется. Оценивайте реально свои возможности и, если у вас появились другие интересы, то в этом нет ничего страшного. Нужный проект выживет. Свободные исходники никуда не исчезают.

0.1 alpha, 23 июня 2000

Первая самостоятельная версия. Наиболее важное из изменений было сделано под впечатлением подробного описания xkb от Ивана Паскаля (**http://pascal.tsu.ru/other/xkb/**).

0.2 все ещё alpha, 30 июня 2000

Безусловно, выпуск был посвящён замечательному ресурсу от Алексея Канавина RU.LINUX Frequently Asked Questions (**http://www.sensi.org/~ak/linuxfaq/rulinux.faq.html**).

Был существенно переделан раздел по кириллизации *LaTeX*. Это время, когда появилась стандартная кириллизация (именно кириллизация, а не русификация) *LaTeX* прямо из коробки. Дистрибутив *tetex* версии 1, содержащий шрифты от Ольги Лапко, вышел в июне 1999 года. До этого для работы с *LaTeX* приходилось использовать сторонние русификаторы.

0.3 ещё alpha, 6 августа 2000

ChangeLog: Сменил имя Cyrillic HOWTO на Cyrillic HOWTO (rus). Название стало теперь соответствовать содержанию, так как с подачи Антона Зиновьева (Болгария!!!) были сделаны добавления не только про русский язык, но и про кириллицу вообще.

0.5 почти beta, 13 сентября 2000

ChangeLog: Всё таки личность в истории имеет значение. Антон Зиновьев переделал около двадцати процентов всего HOWTO. Раздел про консоль – это его раздел.

Кстати, свою собственную нумерацию я начал в частности по той причине, что ощутил силу Linux-сообщества. Приятно, когда тебе говорят спасибо и предлагают исправления.

0.6 уже beta, 14 октябрь 2000

ChangeLog: Хотелось бы отдать должное Андрею Чернову – крёстному отцу koi8 (**http://koi8.pp.ru/main.html**).

Споры, какая кодировка лучше, возникают до сих пор. Смысл их немного не понятен. Ясно, что все мы будем пользоваться Unicode, но пока это светлое будущее не настало, мне достаточно koi8-r.

Сюда же вошли и исправления от Алекса Отта (**http://xtalk.msk.su/~ott/**) – человека, который перевёл документацию по emacs. Благодаря его переводу я и освоил этот текстовый редактор.

0.9 пусть будет beta, 5 ноября 2000

ChangeLog: Мой друг, прочитав этот текст долго смеялся. Когда я спросил: «Что здесь смешного?» – он сказал, что давно не видел такого количества ошибок и издевательств над русским языком. Сжалившись над моей безграмотностью он поправил некоторые из ошибок, но попросил не называть своё имя – скромный очень.

Людей делает язык, поэтому говорить и писать надо правильно. Сказать легко, но реальность такова, что любые тексты надо завершать словами: «Критику и предложения принимаю с благодарностью». Любые исправления и правки – ценность.

ChangeLog: Сильно поменял раздел Emacs и XEmacs, добавил раздел MULE – подробную информацию предоставил Александр Михаилян (автор Belarusian-HOWTO).

ChangeLog: По мотивам FAQ fido7.ru.tex от Михаила Колодина (**http://cyrtex.da.ru/**) изменил раздел про LaTeX. Теперь в тексте появилась информация не только о том как использовать кириллицу, но и как это делать правильно «по русски».

0.99 почти stable – надоело уже все бета, да бета, 12 августа 2001

За полгода накопилось исправлений от двух десятков человек. Документ становится популярным. Это определённо напрягает. Если бы wiki (**http://ru.wikipedia.org/wiki/Wiki**) возникло тогда, то HOWTO разви-вался бы и сейчас. Необходимо распределять нагрузку.

0.999 до stable ещё пилить и пилить, 25 августа 2001

ChangeLog: Настроил psgml (мода в emacs для sgml) – однако, Emacs стал задумчивым. До этого момента я думал, что Celeron 550 для набора текста – это вполне достаточно.

Появился словарь ispell от Александра Лебедева (**http://semiconductors.phys.**

msu.su/~swan/orthography.html) с поддержкой буквы «ё». Записываюсь в фан-клуб любителей буквы под впечатлением текста В. Т. Чумаков «Ё – седьмая, счастливая буква азбуки» (**http://www.gramota.ru/mag_arch.html?id=127**).

Возникло желание включить HOWTO в LDP (Linux Document Project). В LDP отослали в русское отделение. Русское отделение мертво. Мдаа.

Версия 1.0, 17 августа 2002

Прошёл год, HOWTO созрел до стабильной версии.

Версия 1.1, 25 августа 2002

ChangeLog: Перевод документа на другой базовый формат XML. Linuxdoc, похоже, скончался или близок к этому состоянию.

Ретроспективно я теперь понимаю, что это была стратегическая ошибка. От добра-добра не ищут и формат на полпути не меняют, тем более на XML. Позже наткнулся на сайт ненавистников XML (**http://xmlsucks.org/**) – подписываюсь под критикой.

Версия 1.1.2, 7 сентября 2002

ChangeLog: Разбил документ на части, так как Emacs прямо надрывается и без слёз смотреть невозможно (здесь стоит грустный смайлик).

Последствия неверного технического решения. XML просто не предназначен для редактирования руками.

ChangeLog: Сменил название с Cyrillic HOWTO (rus) на Cyrillic HOWTO (russian).

Версия 1.2.3, 21 сентября 2002

Вместо дополнений и исправлений воюю с XML.

ChangeLog: Добавил раздел Глава 15. Кириллизация дистрибутивов.

Надо сказать, что в HOWTO я пытался избежать зависимости от типа и версии дистрибутивов. Но народ хочет не только теории (зрелищ), но и инструкций (хлеба). Пришлось пойти на попятный. Как тогда не хватало wiki.

Версия 1.2.3, 21 сентября 2002

Добавлено индексирование. Документ довольно здоровый. Интересно, хоть кому-нибудь помогло?

Версия 1.4.2, 29 сентября 2002

Добрался до исправлений и предложений. Около десятка изменений. «Проявились» украинцы.

Версия 2.0, 20 ноября 2002

ChangeLog: Благодаря Мише Брукману был добавлен раздел на английском языке для тех, кто не имеет возможности читать на русском, но хочет прочитать этот документ. Гип, гип ура: раздел Chapter 1.

Ещё одна попытка включиться в LDP. Ох уж это желание официального признания. Та же история: отсылка к русскому подразделению, а оно, похоже, и не существовало. Не хватает структуры. Вон у японцев MULE для emacs спонсирует министерство экономики, торговли и производства (**http://www.m17n.org/**). Завидно.

Версия 2.2, 5 октября 2003

Работа по инерции: внесение исправлений от сообщества. XML раздражает всё больше и больше. Слишком много я о нём узнал – издалека эта технология выглядела гораздо привлекательней.

Версия 2.3, 6 октября 2003

ChangeLog: Сменил лицензию на GFDL, добавил раздел GNU Free Documentation License от Елены Тяпкиной (**http://www.libertarium.ru/libertarium/etiapkina**).

Выбор лицензии определён позицией FSF (Free Software Foundation) **http://www.gnu.org/licenses/licenses.html#FDL**. В тексте HOWTO отсутствуют инвариантные секции, поэтому, даже с точки зрения политики сообщества Debian, этот документ свободен.

Версия 2.4, 9 октября 2003

Последние правки и прекращение работы над документом.

Причины прекращения: неожиданно выяснилось, что мне этот проект больше не нужен. Alt Linux, стоящий у меня на домашнем компьютере к тому времени уже больше года, был нормально кириллизован сразу из коробки.

Ноябрь 2005

Перешёл с Alt на Debian. Никаких проблем: кириллизация из коробки это норма. Победа? Скорее начало нормальной жизни.

В текущих комментариях к версиям я выпустил фамилии многих из тех кто мне помогал и благодаря кому Cyrillic HOWTO (russian) стал полезным для сообщества. Я сфокусировался на важных с моей точки зрения событиях, хотя каждое исправление и предложение было ценно. Имена всех перечислены в самом тексте HOWTO. Последнюю версию документа можно взять, например, на моём сайте **http://www.inp.nsk.su/~baldin**. Это открытые исходники и Вы можете изменить их, если посчитаете это нужным Вам или сообществу.

ПРОГНОЗ НА БУДУЩЕЕ

Будущее предсказать не возможно, но никто не мешает попробовать. Здесь я отмечу четыре момента, которые могут существенно повлиять на кириллизацию окружения GNU/Linux, как, собственно, и на сам GNU/Linux.

Unicode – выхода нет

В ближайшем будущем нам грозит Unicode. В условиях глобализации и необходимости в многоязычных документах достойной альтернативы не наблюдается. Нам хуже не будет, ведь мы привыкли к разнообразию кодовых страниц. Вот каково будет англоязычным разбираться с различными представлениями Unicode?

Проблемы только начинаются. Есть масса программ без достойных альтернатив, которым потребуется глобальная переделка. В любом случае надо действовать из соображений, что «лучшее враг хорошего».

wiki – распределённая работа

Эффективность распределённой работы доказана самим существованием GNU/Linux. Недавняя игрушка wiki (http://ru.wikipedia.org/wiki/Wiki) имеет все шансы стать основным инструментом создания качественной открытой многоязычной документации. Простота внесения необходимых исправлений завораживает. Безусловно, это качественно новый уровень.

С моей точки зрения, это – наиболее подходящий кандидат для создания идеального инструмента переводов сообщений программ и документации. Этот путь, каким кириллизация «всего и вся» должна следовать.

На текущщ момент основное преимущество традиционных текстов перед «вики» заключается в возможности легко сделать из текста твёрдую копию. Традиционные тексты в большинстве своём ориентированы на печать. «Вики» следует научиться переводить содержимое в текстовый формат, который при минимальных изменениях позволит делать книги. Мне представляется, что связка LaTeX плюс pdf здесь выглядела бы весьма органично. Тем более что с LaTeX-нотацией «вики» уже знакома (**http://en.wikipedia.org/wiki/Help:Formula**).

Дисплеи – технический прогресс

В отличии от, скажем, винчестеров дисплеи не очень-то и прогрессируют. До сих пор 72 dpi (72 точки на дюйм) считается нормальным разрешением. Увеличение разрешения в четыре раза (до 300 dpi) даст возможность использовать на экране те же шрифты, что и при печати. Растровые шрифты уйдут в прошлое.

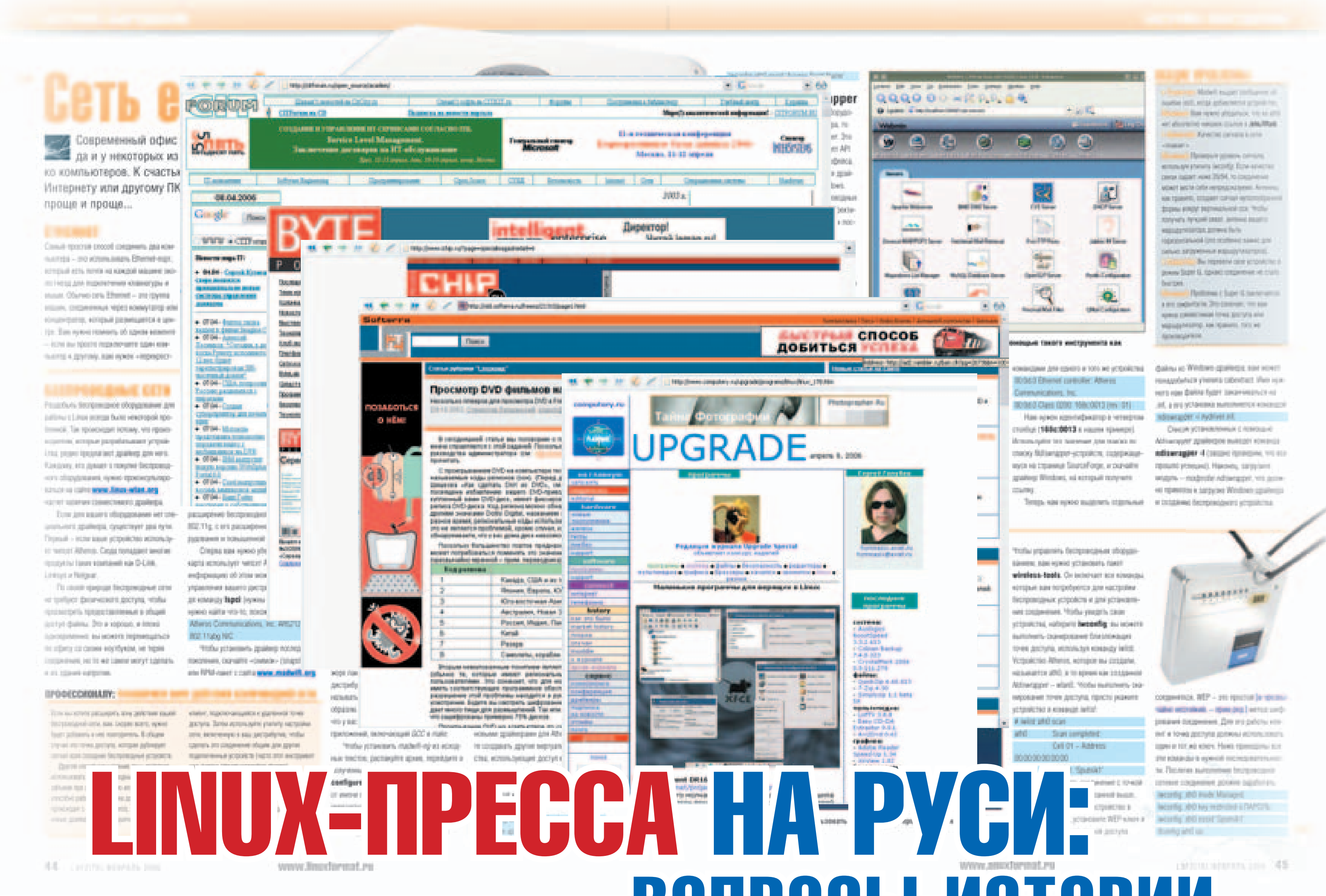
Возможность писать по русски целиком определяется наличием необходимых шрифтов. Хороший шрифт – редкость. Создание шрифтов lh, а затем и cm-super закрыло проблему кириллицы в LaTeX. Чтобы использовать их и на экране, необходим технический прогресс.

Китайцы идут

В своё время мне на глаза попалась написанная с большой любовью и знанием дела зарисовка на тему: парад на красной площади в честь «Юбилея кириллического письма». Там в шуточной форме в виде оформления праздничных колонн были продемонстрированы «компьютерные проблемы» кириллицы. Замыкал же описанный парад «духовой оркестр братских японо-китайских народов, несущих огромный букет свежих белых хризантем, символизирующих искреннее чувство белой зависти к не понимающим своего счастья русскоговорящим программистам и пользователям…»

В силу того, что выхода особого у них нет, и японцы, и китайцы в будущем внесут определяющий вклад в развитие многоязычия в Linux-среде. А там где будут работать и отображаться иероглифы – найдётся место и кириллическим буквам. Необходимо отслеживать полезные начинания в этих регионах.

Говорить и писать на своём языке – это наша судьба. Хорошо, когда этому ничего не мешает.



LINUX-ПРЕССА НА РУСИ: ВОПРОСЫ ИСТОРИИ

О том, как это было, рассказывает **Алексей Федорчук**

На этих страницах я хотел бы проследить, как менялось освещение Linux на страницах печатной компьютерной периодики. Здесь не будет ни полной библиографии статей, ни даже упоминаний всех изданий, уделивших место для предмета нашего разговора. Моя цель, скорее, выявить тенденции развития Linux-прессы в прошлом и ее перспективы – в грядущем. Термин «Linux-пресса» употребляется здесь для краткости – не следует забывать, что речь идет также о Unix и Open Source вообще.

Но сначала – несколько слов о том, с чего все началось. У истоков Linux-прессы лежал журнал «Монитор» (вскоре прекративший свое существование), который в 1994 году напечатал статью Владимира Водолазкого о том, как легко и без головной боли установить Linux. В этой статье, вероятно, слово Linux впервые прозвучало в русскоязычном окружении. Актуальность же она сохраняла еще годы спустя, будучи не только источником информации об этой ОС, но и руководством к действию: ксерокопии ее ходили по рукам в сопровождении горы дискет с дистрибутивом Slackware, и, вероятно, не одно поколение «линуксоидов» ставило свою первую систему с этого «комплекта». Итогом той давней публикации стала книга Владимира «Путь к Linux» – первое (1999 год) отечественное издание на эту тему. Но это – уже другая история.

Первым журналом, в котором сложилась устойчивая традиция линуксописания, не прерывающаяся и по сей день, стал «Мир ПК». В нем, начиная с 1995 года, регулярно начали публиковаться как переводные, так и оригинальные статьи о самых разных аспектах ОС Linux. Причем отечественные авторы преобладали, и их публикации проявили отчетливую тенденцию к циклизации: до сих пор памятны циклы статей Игоря Хименко (по совместительству – одного из соратников Сергея Кубушина, развивавшего тогда на Украине инновационный дистрибутив KSI; однако и это отдельная тема), посвященные файловой системе Linux, процессам, командной оболочке bash. Правда, со временем традиция циклических публикаций в «Мир ПК» угасла…

Начиная с 1997 года, многие общекомпьютерные журналы начинают публиковать статьи по тематике Unix, Linux и Open Source – не часто, но относительно регулярно такие материалы появляются в журналах «PC Magazine/RE», «LAN» и «Открытые системы». Характер этих публикаций различен. Если «PC Magazine/RE» печатает почти исключительно переводные статьи весьма случайного содержания, то «LAN» и «Открытые системы» (тот же издательский дом «Открытые системы», что и «Мир ПК») отдают предпочтение отечественным авторам, и в их материалах можно наметить ту же тенденцию к циклизации, впрочем, также не получившую развития. Упомяну тут и журнал «СУБД», проживший короткую, но яркую жизнь (оборвавшуюся после приснопамятного августа 1998 года) – это был практически единственный компьютерный журнал «академического» стиля. И, хотя специальных материалов о Linux в нем не было, его тематика тесно пересекалась с Unix и Open Source.

В общем, разнородность содержания и отсутствие систематического подхода к Linux-тематике характерна для большинства общекомпьютерных изданий и по сей день. Но были и попытки целенаправленного освещения Unix, Linux и Open Source.

Первым в этом ряду должно упомянуть журнал Byte Россия. Начав свое существование в 1998 году под эгидой издательского дома «Питер», он с первых же номеров начал регулярно печатать материалы по Unix и Open Source, причем

переводные статьи сразу же составляли меньшинство по сравнению с работами отечественных авторов. Очень скоро эти материалы были выделены в специальный раздел – Byte/Unix, своего рода журнал в журнале, редактором которого стал Алексей Выскубов. Это была первая попытка создания специализированного издания по тематике Unix, Linux и Open Source вообще: помимо статей о Linux, здесь уделялось внимание и BSD-системам, и общим вопросам идеологии свободного программного обеспечения. В частности, именно на его страницах увидели свет переводы статей Николая Безрукова об особенностях разработки программ с открытыми исходными текстами, сохраняющие актуальность и по сей день (благо, они доступны в сети, например: **http://citforum.ru/open_source/academ** и **http://citforum.ru/open_source/catb_sec_view**).

Вообще трудно переоценить роль Byte Россия в развитии русской Linux-прессы того времени – тираж его в 2000 году достиг 17 тысяч, и не последнюю роль в этом сыграли материалы Unix-раздела. Однако, в конце 2000 года журнал был продан другому издательскому дому, политика его резко изменилась, раздел Byte/Unix был ликвидирован, полностью сменилась редакционная команда. И хотя в начале 2001 года в нем еще по инерции публиковались некоторые материалы по Linux, интерес к нему со стороны линуксописателей (и, подозреваю, линуксочитателей) был утрачен безвозвратно.

Однако развитие Linux-прессы продолжалось, правда, существенно сместившись в сторону онлайновых СМИ. Эстафету подхватила Компьютерра: в начале 2001 года в рамках этого издательского дома возник проект Софтерра (редактор Сергей Scout Кащавцев) со специальным разделом – FreeOS (Федор Сорекс), посвященным свободному ПО вообще и ОС Linux (а также Free-, Open- и прочим BSD), в частности. Все материалы проекта публиковалась в Сети, но некоторые статьи раздела попадали также и на страницы «бумажной» Компьютерры.

Наибольшим успехом проекта можно считать выход тематического номера журнала «Домашний компьютер», тоже относящегося к издательскому дому Компьютерра (ДК, 2002, #12). Созданный под идейным руководством Максима Отставнова, он содержал материалы по всем аспектам устройства и использования Linux, в том числе и в домашних условиях. К сожалению, этот успех был последним: в течение первых месяцев следующего года проект FreeOS (да и Софтерра как таковая) плавно сошел на нет, материалы его исчезли из прямого доступа с сайта **http://www.computerra.ru**. Часть их по сей день можно найти по адресу **http://old.softerra.ru**, но кое-что, похоже, утрачено безвозвратно, как и онлайновая версия тематического выпуска ДК.

Следующим номером в эстафетном забеге Linux-прессы оказался Upgrade – усилиями Алены Приказчиковой и Сергея Голубева, начиная с середины 2002 года, почти каждый номер в «софтверном» разделе содержал материалы про Linux и Open Source, чему способствовал и приток новых авторов. Тенденция к систематическому освещению темы нашла свое воплощение в тематическом выпуске Upgrade Special, посвященном Linux вообще и дистрибутивам Live CD в особенности (середина 2004 года). Однако, как и в случае с ДК, он одновременно стал символом упадка интереса к Linux и Open Source со стороны редакции Upgrade: в последующее время материалы по этой тематике появлялись там лишь эпизодически.

Сказанное не значит, что остальная компьютерная периодика перестала уделять внимание Linux-тематике. С самого дня своего создания (начало 2001 года) значительную активность на этом поприще проявлял журнал Chip – вплоть до выпусков спецномеров, целиком посвященных Linux, и сопровождавшихся компактом с тем или иным его дистрибутивом. В 2002 году возник журнал «Системный администратор», коему по титулу положено освещать вопросы, связанные с сетевым администрированием – а в этой сфере Linux и свободные BSD-системы доминируют и по сей день (особенно на территории РФ). И материалы о них появляются на его страницах с завидной регулярностью. Время от времени к тематике Linux и BSD обращается «Хакер» – правда, подчас с материалами весьма спорными.

В ряду общекомпьютерных периоди-

ческих изданий следует выделить упомянутый ранее журнал «Открытые системы». Его специфика в том, что, наряду со статьями популярного направления он публикует и сугубо научные материалы, посвященные теоретическим вопросам Computer Science (что не удивительно, учитывая его академические корни). К сожалению, популярность его все время падает, а в последние годы он практически пропал из розничной продажи.

Журналы, публикующие материалы по нашей тематике лишь эпизодически, на общую картину Linux-прессы влияли слабо. Ее путь, со дня зарождения и почти до сегодняшнего момента, можно описать как серию попыток создать специализированное издание, профилированное на Unix, Linux и Open Source. Ни одно из этих предприятий успехом не увенчалось. Причины неудач можно выискать разные, но в основе каждый раз лежало одно: абсолютное равнодушие руководства общекомпьютерных изданий (даже таких, как «Открытые системы», которых, казалось бы, даже титул обязывал) к Linux-тематике. Дело помощи линуксоидам следовало брать в руки самих линуксоидов.

Таким образом, мы плавно подошли к 2005 году. Я не пророк, но думаю, что он войдет в историю как второй, после 1998 года, переломный рубеж в развитии Linux в России. Если первый ознаменовался выходом прототипа первого российского дистрибутива – Mandrake Linux от IPLabs Linux Team (в последующем – Altlinux) и попыткой создания первого специализированного издания (Byte/Unix), то в течении 2005 года, во-первых, в России резко активизировалась деятельность лидеров мирового дистростроения (Red Hat и Novell/Suse), и, во-вторых, к нему приурочены первые массовые выставки-конференции, специально посвященные тематике Open Source (Open Source Forum Russia, LinuxWorld Russia, LinuxLand на Софтуле).

А для Linux-прессы год этот памятен тем, что в течении его начали выходить первые специализированные периодические издания, полностью посвященные Linux и Open Source – Chip Linux Special и Linux Format.

Chip Linux Special – ежеквартальное издание, которое берет свое начало с весны 2005 года. Он генетически связан со специальными выпусками журнала Chip, и издается тем же издательским домом «Бурда», что и прародитель. Комплектуется исключительно оригинальными статьями отечественных авторов. Специфика журнала – сконцентрированность вокруг материалов по «титульной» ОС, расширение тематики в сторону других свободных систем (например, BSD-семейства), насколько я знаю, не планируется.

Эта статья в целом уже была написана, когда поступила печальная весть: журнал Chip Linux Special более издаваться не будет. О причинах этого мне ничего не известно, да и сами источники сложно назвать официальными: помимо личного общения с членами команды Chip Special Linux можно назвать только новость на **Linux.org.ru**, в которой (со ссылкой на телефонный разговор с издательским домом «Бурда») сообщалось, что журнал было решено закрыть. Хочется надеяться, что «слухи о его смерти сильно преувеличены» и Chip Special Linux еще порадует нас своими выпусками – благо приобрести его можно было «от Москвы и до Находки».

Linux Format – ежемесячный журнал, родившийся в сентябре 2005 года усилиями фирмы Линуксцентр, занимавшейся онлайновой торговлей дистрибутивами свободных ОС и профильной литературой. Он представляет собой перевод одноименного английского журнала, своего рода русское его зеркало: каждый номер по содержанию соответствует аналогичному номеру оригинала (хотя и выходит с некоторым запозданием, обусловленным

затратами времени на перевод, верстку и печать). За одним исключением: начиная с 4-го номера, в журнал включаются и оригинальные статьи отечественных авторов. Рискну предположить из исторических аналогий, что с течением времени процент последних будет возрастать. Ведь все патриархи отечественной компьютерной прессы (Мир ПК, Компьютер-Пресс, PC Magzine/RE) начинались некогда как чисто переводные издания.

Отличительная черта Linux Format – открытость. Все стороны его существования – от содержания очередного номера до причин недоставки конкретного экземпляра – можно обсудить в специальной теме Линуксфорума (**http://www.linuxforum.ru**). Члены редакционной команды в обсуждении участвуют – и, должен вам заметить, к обоснованным мнениям читателей всегда прислушиваются.

Впрочем, много говорить об этом издании не буду: раз вы читаете материал этого номера, то знакомы с ним в достаточной степени. А потому завершу свое сочинение рассуждениями на тему, каким видится идеальный журнал по тематике Unix, Linux и Open Source (см. врезку).

НАШИ СОБЫТИЯ: ЧТО БЫЛО, ЧТО БУДЕТ

30 марта 2006 года в Международном информационно – выставочном центре «ИнфоПространство» прошла конференция «Решения On Demand». Организатором конференции выступила компания «Теллур – Сетевые Технологии» (www.tellur.ru) при поддержке компаний IBM (www.ibm.ru/com), APC (www.apc.ru) и Landata (www.landata.ru). Информационными спонсорами мероприятия выступили компании Linuxcenter (www.linuxcenter.ru) и журнал «Linux Format» (www.linuxformat.ru).

На конференции присутствовали представители более 30 компаний из России и ближнего зарубежья.

В рамках конференции были представлены новинки серверного оборудования IBM линейки pSeries, рассмотрены системы хранения Total Storage, а также Адаптивная Инженерная Инфраструктура центров обработки данных (NCPI) компании APC.

Центральное место в конференции заняла демонстрация возможностей виртуализации, как эффективного способа оптимизации работы сервера, проведенная генеральным директором компании «Теллур – Сетевые Технологии» Дмитрием Лобановым. Слушателям была предоставлена возможность наглядно убедиться в достоверности сенсационных заявлений компании IBM о высокой готовности и производительности серверов линейки pSeries с RISC архитектурой под управлением ОС Linux.

Особый интерес у слушателей вызвал доклад Руководителя Центра компетенции Linux Дениса Сосновцева, в котором были освещены ключевые аспекты работы под ОС Linux, рассказано о преимуществах и особенностях данной ОС, решениях, построенных с ее помощью и продуктах, работающих с ней.

Также со ссылкой на официальный источник было объявлено о планирующемся выходе бета-версии программы 1С, работающей под операционной системой Linux, с базой данных, построенной на PostgreeSQL сервере.

В апреле 2006 г. стартовал РИФ-Марафон – уникальный всероссийский проект, целью которого является популяризация интернет-технологий в российских регионах, обмен опытом в ИТ-сфере на региональном уровне, подготовка региональной программы РИФ-2007, а также обеспечение базиса для Фестиваля региональных интернет-проектов «Новая реальность».

РИФ-Марафон представляет собой серию выездных мероприятий (семинары РОЦИТ, круглые столы, мастер-классы и презентации) и региональных спец.мероприятий в рамках подготовки Фестиваля «Новая реальность», которые пройдут в 7 федеральных округах РФ.

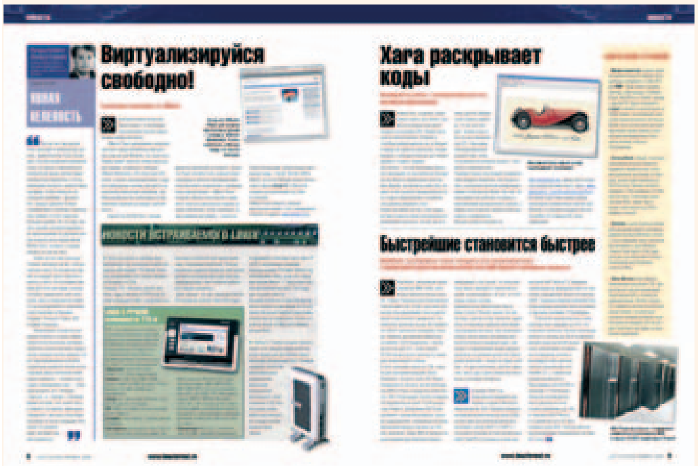
В ходе научно-практической и дискуссионной части таких мероприятий в регионах будут обсуждаться самые острые проблемы Рунета, затронутые во время РИФ-2006 и признанные наиболее интересными для региональных участников Форума.

Одним из важных акцентов РИФ-Марафона станет обсуждение регионального аспекта развития интернет-технологий и возможностей обмена опытом на региональном уровне.

ИДЕАЛЬНЫЙ ЖУРНАЛ

Традиционно «толстые» компьютерные журналы разделяются на две части: блок новостей и, так сказать, «тело» журнала – собственно материалы номера. Оправдана ли такая организация в век тотальной интернетизации? В век, когда все, имеющие хоть какое-то подключение к Сети, получают интересующие их последние известия из онлайновых источников, новостные разделы даже компьютерных еженедельников выглядят сборником анекдотов с бородой Карла Маркса. Что же тогда говорить о «новостях» ежемесячников?

Так что же, ликвидировать новостные блоки? Отнюдь – это было бы политически неправильным. Увы – изрядная часть населения постсоветских пространств лишена прелестей Интернета, и Печатное Слово, пусть несколько устаревшее (да и доставленное, силами российской почты,



с запозданием), для нее – единственный источник информации. А потому предлагается компромиссный вариант: заменить сборники анек... пардон, новостей – аналитическими их обзорами. Которые, давая достаточно сведений читателям, не имеющим подключения к Сети, в то же время не вызывали бы раздражения своей «бородатостью» у тех, кто таковое имеет. А в идеале – были бы просто интересны сами по себе. Конечно, составление таких обзоров – дело нелегкое, но оправдается повышением читательского внимания.

Теперь об основной части – статьях. Журнал, рассчитывающий на самоокупаемость (а в идеале – и на принесение прибыли) обязан ориентироваться на самые широкие пользовательские массы И потому должен содержать материалы нескольких градаций: для совсем начинающих, для «действующих» пользователей, и для тех, кто ставит своей задачей углубленное изучение каких-либо частных вопросов. Хорошо это или плохо – обсуждать не будем, такое «смешение жанров» на данном этапе развития Linux-прессы является необходимостью. Как показала трагическая кончина журнала «СУБД» (да и безрадостная судьба аккумулировавших его «Открытых систем»), специализированное издание «для профи» пока не имеет шансов выжить на постсоветском пространстве. С другой стороны, ориентация издания на «чайников» чревата потерей интереса к нему, как только «чайники» таковыми быть перестанут (а с помощью хорошего журнала это произойдет очень быстро).

Какой видится компоновка материалов? Возможны варианты: по степени «продвинутости» предполагаемого читателя, по тематике, в том числе и с выделением некоего центрального материала и его «системного окружения». Впрочем, я – категорический противник «темы номера», что приемлемо для еженедельника, но для ежемесячного издания смертельно: ведь если читателю не интересна именно эта тема номера, он на целый месяц лишается возможности что-либо почерпнуть из любимого журнала.

Формат большинства журналов общего назначения предполагает преимущественно двух- или, реже, четырехполосные статьи, что для глубокого изложения многих животрепещущих проблем явно недостаточно. Конечно, проблема эта для

периодических изданий не решаема в принципе: увеличение объема статей повлечет за собой сужение тематики номера и риск утраты читательских симпатий. Однако некий компромисс тут возможен – в виде пролонгированных из номера в номер тематических циклов.

Для журнала тематики Unix и Linux очень существенен общий дизайн – и здесь положение, в большинстве случаев, не удовлетворительное. Многоколоночная верстка, пришедшая из мира рекламно-развлекательной периодики, и терпимая в периодике, так сказать, литературно-повествовательной, оказывается проклятием в изданиях технического профиля. В нашем случае это относится в первую голову к командам и листингам. Писать без них серьезно про Linux и Unix – это все равно, что писать про музыку без нот, или про живопись – без репродукций. Но – увы – длинные команды или содержимое конфигурационных файлов вписывается в облик страницы традиционного современного журнала ничуть не лучше, чем «парень в джинсах и кожаной куртке» – в интерьер ресторана для новорусского истеблишмента.

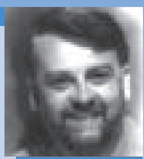
Я прекрасно понимаю, что общий дизайн издания определяется множеством приходящих факторов (в том числе и политических). Но и тут возможны варианты. Например, давать необходимые команды и листинги «внеформатными» врезками. Или – сделать онлайновое дополнение к журналу, которое содержало бы именно ту часть статей, которая подлежит использованию методом Cut&Paste. Такое дополнение, не дублируя содержание основного материала, будет способствовать его практическому использованию.

Вопрос? Ответ!

Если вы завязли в какой-то проблеме и чтение HOWTO не помогает, почему бы не написать нам? Наши эксперты помогут вам разобраться даже в самых сложных проблемах.

НАШИ ЭКСПЕРТЫ

Наши эксперты найдут ответ на самый трудный ваш вопрос. Если у вас проблемы с установкой, настройкой модема, сетью или еще чем-нибудь – просто напишите нам, а обо всем остальном позаботимся мы.

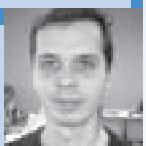


Управляя Интернет-провайдером, а заодно подрабатывая редактором дисков *LXF*, **Нейл Ботвик** (Neil Bothwick) скромно зовет себя мастером на все руки.

Брэндон Калигари (Brendon Caligari) больше десяти лет работает с Linux, он администратор работающей системы в фирме Rackspace Managed Hosting.

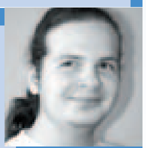
Сергей Владимирович Рыжиков

Генеральный директор компании «БИтрикс»



Максим Владимирович Смирнов

Ведущий программист отдела разработки компании «БИтрикс»



Александр К. – сторонник Unix-way. Молодой, но перспективный член дружной команды экспертов.

Валентин Синицын Поддерживает проект Slackware Reiser4, интересуется настольными Linux-технологиями и рад помочь Вам разобраться с ними.

Безbashенному

Я пытаюсь писать *bash*-скрипты с использованием утилит, требующих ввода с клавиатуры. Например, здесь без этого не обойтись:

```
update-alternatives --config xxx
```

Хотелось бы автоматизировать этот процесс с помощью параметра, передаваемого скрипту. На данный момент мое лучшее решение – записать в файл то, что нужно ввести с клавиатуры, запустить **update-alternatives**, направив поток ввода из этого файла, а затем удалить сам файл. Но должен быть другой путь. Как можно ввести параметр с клавиатуры, не записывая его во временный файл?

Аноним, с форума LXF

Запомните закон Linux (и Unix): «Все на свете – файлы», включая стандартные потоки ввода и вывода. Для них предусмотрены дескрипторы: **&0** для стандартного ввода *stdin*, а **&1** – для стандартного вывода *stdout* (кстати, **&2** – это потока ошибок, *stderr*) . Вот что Вам нужно:

```
echo «A» | update-alternatives --config xxx <&0
```

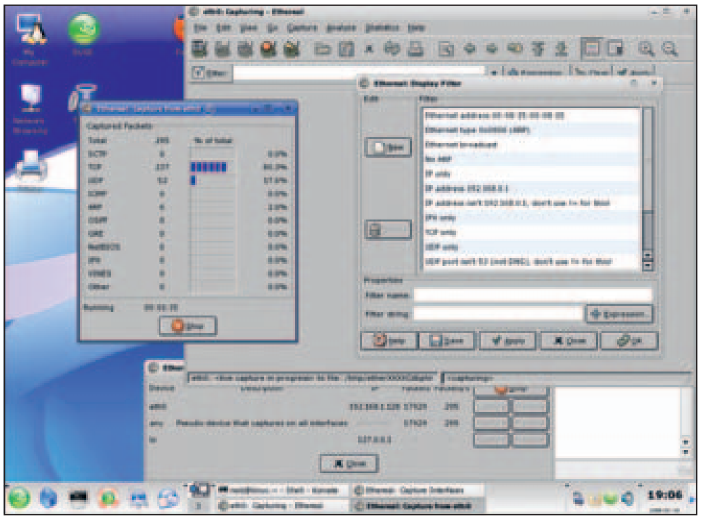
Здесь «A» – параметр, вводимый с клавиатуры. *echo* посылает его в *stdout*, далее канал (|) перенаправляет его следующей команде: **&0** – дескриптор для stdin, а **<&0** направляет ввод на команду.

Согласно другому неписаному закону Linux, «для любой задачи существуют как минимум два способа выполнения». Вместо **&0**, **&1** и **&2** Вы можете указать **/dev/stdin**, **/dev/stdout** и **/dev/stderr** соответственно. Версии с **&** короче набирать, но **/dev** будут понятнее, если Вы откроете свой скрипт через полгодика. ***НБ***

Вдохнуть Ethereal

Только что приобрел ваш журнал и установил SUSE Linux на списанный компьютер, потому что очень хочу разобраться в Linux. Установка была очень простой, и руководства в журнале очень полезные. Однако я сетевой инженер, поэтому теперь хочу установить *Ethereal* [анализатор сети]. Поскольку я новичок в Linux и у меня очень мало опыта, не могли бы вы подсказать, что нужно скачать, чтобы скомпилировать и установить *Ethereal* в SUSE Linux?

Линден Данлоп



Большинство дистрибутивов поставляется с кучей готовых пакетов, поэтому компилировать программы вроде *Ethereal* вряд ли придется – разве что понадобилась самая свежая версия.

Собрать из исходных текстов любую открытую программу очень легко. Однако у дистрибутивов вроде SUSE Linux есть замечательное свойствО большинство необходимого имеется на дисках или в репозитариях пакетов.

Для установки *Ethereal* запустите *YaST* из меню SUSE, перейдите на вкладку Программное обеспечение (Software) и выберите Управление программным обеспечением (Software Management). Теперь просто напечатайте «ethereal» в строке поиска, выберите найденные пакеты и нажмите Принять (Accept) для запуска установки. Если у программы есть какие-нибудь зависимости (другие программы или библиотеки), они установятся автоматически.

По умолчанию *YaST* знает лишь о тех пакетах, которые есть на установочных дисках. Вы можете добавить репозитарии с дополнительными пакетами, выбрав Источники установки (Installation Sources). Список зеркал SUSE можно найти на сайте **www.opensuse.org/Mirrors_Released_Version**. Выберите себе сервер и добавьте его в *YaST*, чтобы иметь доступ к самым свежим версиям пакетов. ***НБ***

Уловка-22

Я уезжаю работать за границу на несколько месяцев и хотел бы иметь удаленный доступ к своей сети. Установил *FreeNX* на SUSE 9.3 и перенаправил пакеты с порта 22 на нужную машину на роутере Netgear. Теперь я без проблем могу соединяться со своей машиной, просматривать почту и запускать приложения.

Этим утром я просмотрел **/var/log/messages** и увидел, что кто-то пытается атаковать меня по 22-му порту. Там были сотни сообщений от sshd вида `Invalid user <xxx> from :: ffff:195.90.196.20'.

На моей системе зарегистрированы только 2 пользователя, root и я (вряд ли хакеры могут угадать мое имя). Я использую сильные пароли, комбинируя буквы верхнего и нижнего регистра с цифрами, их нет ни в каком словаре.

Беспокоиться ли насчет этих атак? Можно ли предупредить *sshd*, чтобы он отклонял соединения после X неудачных попыток входа за N секунд, или лучше просто отследить злоумышленника и отклонять вообще все пакеты с его ip-адреса?

Лео Палваст

Такие атаки – обычное дело, если Вы открываете миру порт 22 для внешнего ssh-доступа. Есть несколько приемов, умень-

шающих вероятность проникновения злоумышленника. Сложные пароли – только первый шаг. Так как Вы используете SSH для удаленной пользовательской работы, root-доступ Вам скорее всего не нуже**н, вот и отключите его в** /etc/ssh/sshd_config. Для этого найдите строку

```
PermitRootLogin yes
```

и замените **yes** на **no**. Если понадобится, сможете получить права суперпользователя через **su**, а вот взломщику для этого придется взломать уже два пароля – сначала Ваш пользовательский, затем пароль суперпользователя.

Или включите yes для **without-password**. root-доступ откроется, но только для того, у кого верный ssh-ключ. Посмотрите man-страницы для *ssh* и *ssh-keygen*, чтобы побольше узнать об этом. Вы можете потребовать проверки ключа для всех пользователей, но тогда придется скопировать ключ на все компьютеры, с которых Вы захотите иметь ssh-доступ – это не здорово. Решение сработает, если Вы заходите по ssh только со своего ноутбука.

Для включения этой опции найдите строку с «PasswordAuthentication» и приведите ее к такому виду:

```
PasswordAuthentication no
```

Еще Вы можете запускать ssh-сервер на каком-нибудь нестандартном порту, изменив строку «Port 22» в **sshd_config** и разрешив этот порт в *nxclient* и *knx*. Все эти методы позволят Вам сделать Вашу систему практически недосягаемой для хакеров.

Есть несколько программ, блокирующих IP-адреса, с которых пытаются подобрать пароли для SSH и других сервисов: **http://breakinguard.sf.net**, **http://daemonshield.sourceforge.net** или **www.csc.liv.ac.uk/~greg/sshdfilter**.

SUSE 9.3 больше не поддерживается, лучше обновитесь до версии 10.0.

Криптографомания

В У нас в офисе – внутренний почтовый сервер на *Dovecot*. Мы бы хотели обеспечить нашим сотрудникам шифрованный доступ к нему, чтобы они могли подключаться к серверу из дома, но беспокоимся о безопасности этого шага. Можете ли вы рассказать о том, как разрешить безопасное соединение через надежные email-протоколы?

Мичил Гир

О Шифрование базовых сервисов совсем не трудно, несмотря на то, что математические аспекты криптографии очень сложны для понимания. Нужно только создать SSL-сертификат и убедиться, что сервер работает с созданными нами сертификатами. Сертификат можно и купить, но вряд ли это будет оправдано в случае исключительно внутренних нужд. Будь это общедоступный сервис, Вам потребовался бы сертификат, распознаваемый большинством распространенных почтовых клиентов.

Для создания сертификата воспользуемся пакетом *OpenSSL* (**www.openssl.org**), который есть в большинстве дистрибутивов. Команда openssl сама по себе малопонятна, но есть простой интерактивный интерфейс, через который можно создать сертификат для Вас. Для этого имеется скрипт **mkcert.sh**, поставляемым вместе с **Dovecot**, но, имея дело с файлами *OpenSSL*, можно изготовить ключи и сертификаты другого вида.

От лица суперпользователя перейдите в директорию **/etc/pki/tls/certs** (**/usr/share/ssl/certs** для SUSE или **/etc/ssl** для Mandriva). Если наберете **make** в командной строке, получите напоминание о том, какие сертификаты будут созданы. Обычно сначала создается ключ, а затем на его основе генерируется сертификат. А вот команда **make dovecot.pem** создаст ключ и сертификат для *Dovecot*, оба в одном файле. Все, что Вам нужно – в ответ на подсказку ввести информацию (значения по умолчанию будут показаны в квадратных скобках). Самое важное из заполняемых полей – Common Name, тут следует вписать домен Вашего почтового сервера, а с остальными разобраться нетрудно.

Теперь перенесем созданный сертификат в директорию, указанную в файле конфигурации *Dovecot* – **/etc/dovecot.conf**, параметры **ssl_cert_file** и **ssl_key_file** – это те самые сертификат и ключ. Просто скопируем файл с этой парой в **/etc/pki/dovecot/dovecot.pem** и в **/etc/pki/dovecot/private/dovecot.pem**, удалив при этом автоматически сгенерированные файлы. Уникальный сертификат готов – наслаждайтесь безопасным сервером *Dovecot*. ***КК***

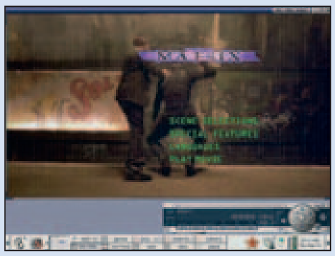
>>

КРАТКАЯ СПРАВКА DVD-CSS

DVD-CSS – это причина, по которой большинстве легальных DVD с фильмами нельзя просмотреть в дистрибутивах Linux «из коробки». CSS предназначен для предотвращения нелегального копирования DVD-дисков путем шифрования записанной информации. Криптографический ключ хранится в специальной области на диске, к которой у стандартных пишущих DVD-приводов нет доступа (для записи). На чистых дисках DVD-/+RW эта область заполнена нулями. При копировании такого диска скопируется вся информация, кроме ключа.

Взлом шифра возможен путем перебора, поскольку в CSS довольно слабый алгоритм защиты – правда, начало показа придется отложить: «кина не будет» около суток. В итоге алгоритм CSS был взломан тремя людьми. Двое из них предпочли безвестность; третьего, подростка Йона-Леха Йохансена (Jon Lech Johansen), теперь кличут не иначе как Йон-DVD. Троица состряпала DeCSS, позднее включенный в *libdvdcss*. Linux-плейеры могут использовать эту библиотеку для просмотра зашифрованных DVD-дисков.

В большинстве стран DeCSS запрещен, а Йон-DVD даже посидел под арестом, хотя и не осужден, как не осужден никто за использование *libdvdcss*. Однако для создателей дистрибутивов было бы чересчур рискованно включать эту библиотеку в стандартную поставку. Тем не менее библиотеку можно отыскать в неофициальных репозитариях: Penguin Liberation Front [Фронт Освобождения Пингвинов] для Mandriva, Packman для SUSE или FreshRPMs для Fedora Core, или собрать из исходных кодов. Так как она нелегальна, даже и не ищите эту библиотеку на наших дисках.



***Xine*, в компании с *libdvdcss*, крутит шифрованные DVD.**

ЧАСТО ЗАДАВАЕМЫЕ ВОПРОСЫ О РУТКИТАХ

FAQ Звучит безобидно, но я так понял, что руткит – нехорошая штука?

Ох, нехорошая: это комплект утилит, позволяющих взломщику скрывать свое присутствие на взломанной машине, вроде модифицированных версий ls и ps, которые не отображают файлы и процессы взломщика. Они прячутся от глаз, как трояны, чтобы не вызывать подозрений. Свое название руткиты получили потому, что они обычно устанавливаются взломщиком, получившим права root (rootkit, в переводе с английского «набор root», – прим.ред.). Пораженный компьютер на компьютерном жаргоне называют «порутанным» (rooted).

FAQ Какой вред обычно причиняют руткиты?

Сами по себе, небольшой. Их задача – не дать владельцу взломанной машины заподозрить неладное по поводу странных файлов или процессов. Они, как кукушата, заменяют собой утилиты, обычно используемые для мониторинга состояния системы. Этим они отличаются от троянов.

FAQ А это что такое?

Троянские кони – сокращенно, трояны – это деструктивные программы, замаскированные под нечто легальное. Имя происходит от хитрости, используя которую греки заняли город Троя. Отсюда пошла фраза «бойтесь данайцев, дары приносящих, и людей, дарящих взломанные программы». Пример троянского ПО –программы класса Adware, к счастью, не распространенные в Linux.

FAQ Какое отношение это имеет к аудиодискам?

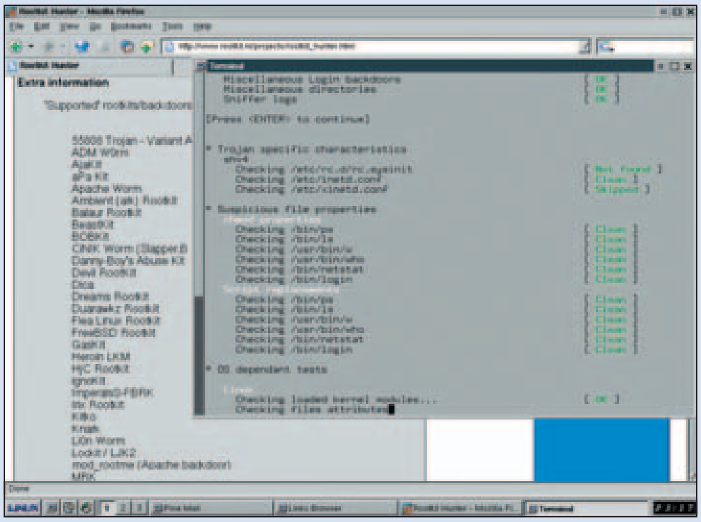
Только то, что неповинная программа Sony DRM была объявлена руткитом, хотя и для Windows. Sony – а еще солидная фирма! – скрытно установила ее на стольких компьютерах без ведома и согласия пользователя, что она стала рекордсменом среди руткитов.

FAQ Как догадаться, что мой компьютер порутан?

Есть программы, выявляющие признаки установки руткита. Их работа подразделяется на два вида: одни ищут сигнатуры знакомых руткитов, другие проверяют MD5-суммы и сравнивают их с контрольными суммами, сделанными в чистой системе, поэтому после установки чистой системы важно как можно раньше дать им проработать. Примеры таких программ – *Rootkit Hunter* (**www.rootkit.nl**) и *chkrootkit* (**www.chkrootkit.org**). Они должны запускаться регулярно по заданию *Cron*. Лучше всего запускать их, когда вы подключены к сети, чтобы они заодно автоматически обновляли базы данных руткитов.

FAQ Эти программы помогут мне восстановить систему?

Нет. Если взломщик установил руткит, целостность системы уже нарушена.



***Rootkit Hunter* может идентифицировать 61 тип руткита.**

Руткит помогает взломщику замести следы, пока он вытворяет в вашей системе все что хочет и пакостит везде. Единственный безопасный путь – регулярное резервирование конфигурационных файлов и полная переустановка системы.

Где модем?

НА ДИСКЕ

После месяцев сомнений я наконец набрался смелости установить SUSE 10. Установка прошла нормально, но система настойчиво считала, что я в локальной сети (хотя у меня никакой сети нет), да еще она не нашла мой модем.

Я испробовал все советы из вашего журнала и все, что нашел в различных форумах, однако без толку (честно говоря, многого из того, что мне советовали, я просто не понял). Я даже хотел удалить Linux с жесткого диска, но теперь уже долгое время я не могу загрузиться в Windows, потому что не запускается какой-то *Grub*.

Если вы сможете мне помочь, я попытаюсь еще раз. Если нет, то как я могу удалить SUSE Linux?

Лес Вуд

Во-первых, важное замечание. Пока Вы не сообщите как можно больше информации о Вашей системе, мы будем блуждать в потемках. Чтобы помочь, мы должны знать конфигурацию компьютера, на который Вы устанавливаете Linux. Есть в Вашем компьютере сетевая карта? Если да, тогда понятно, почему SUSE думал, что Вы в сети. Решение этой проблемы очень простое: запустите *YaST*, зайдите в Сетевое оборудование (Network Devices) > Сетевая карта (Network Card), выделите сетевой интерфейс и нажмите Удалить (Delete).

Теперь займемся модемом. Скорее всего SUSE его не распознает, потому что у Вас так называемый Winmodem. Это примитивные устройства, и большинство их функций перекладывается на центральный процессор и Windows-драйвер. Некоторые из них можно уговорить поработать в Linux, некоторые – нет, поскольку их изготовители не выкладывают спецификаций, которые нужны для создания драйвера.

Для начала определите, что у Вас за модем, с помощью утилиты *Scanmodem*, которая включена на диск этого журнала. Вставьте диск, запустите терминал и выполните такую команду:

```
sh /media/cdrom/Magazine/Answers/
scanModem
```

Скрипт создаст директорию **Modem** с несколькими текстовыми файлами, содержащими информацию о Вашем модеме, о подходящем драйвере и о том, где его найти. Дополнительная информация имеется на сайте **http://linmodems.org**.

С другой стороны, можно приобрести обычный COM-модем (с некоторыми USB-моделями также возникают сложности). COM-модемы без проблем работают абсолютно в любой операционной системе.

Ваша проблема с удалением Linux не уникальна. Когда Вы установили его, в главную загрузочную запись прописался *Grub*, позволяющий выбрать загрузку Linux или Windows. Linux Вы стерли, а *Grub* остался, но необходимые ему файлы были удалены. Поэтому он и не может нормально запуститься. Если у Вас Windows 98, загрузитесь с аварийной дискеты или диска и запустите *fdisk* следующим образом:

```
fdisk /mbr
```

Если у Вас Windows XP, то вместо *fdisk* запустите *fixmbr*. ***НБ***

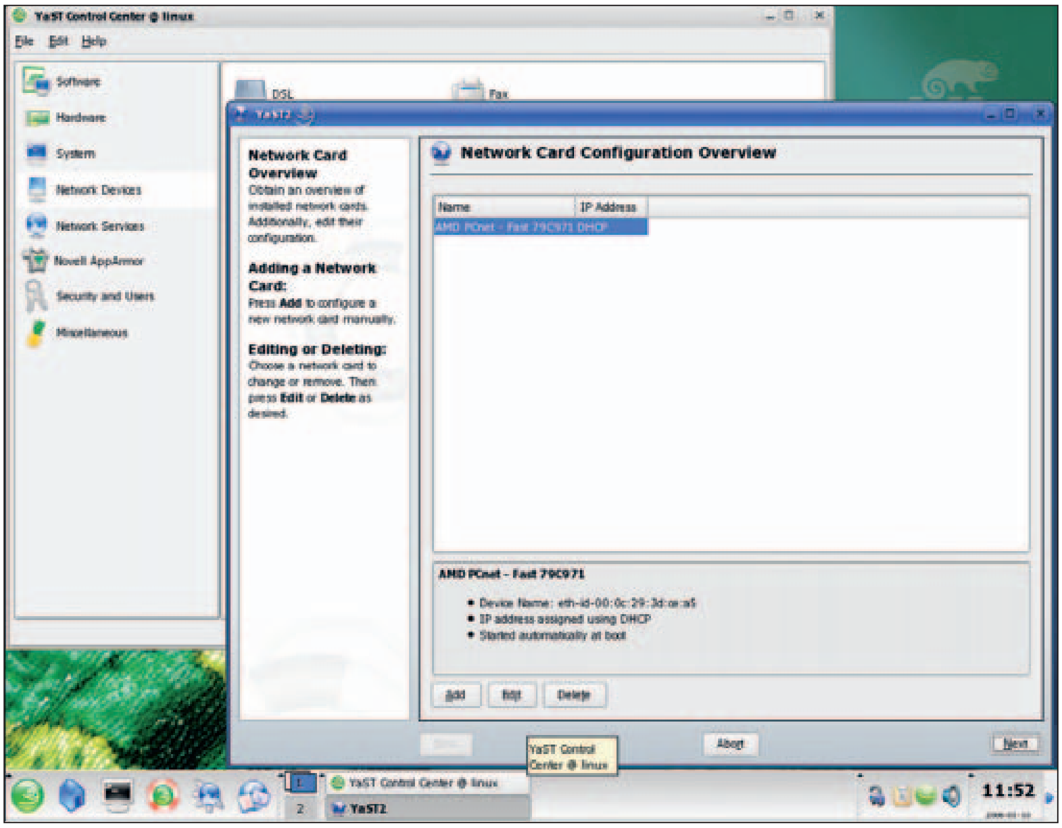
Синхронология

У меня есть старый компьютер с Mandrake 8.2, который служит домашним сервером для шести компьютеров. Мы все еще на ISDN, поскольку в широкополосном Интернете для нас нет насущной необходимости. Последние несколько месяцев я использовал NTP для еженедельной синхронизации времени на сервере. Я создал простое задание в *Cron*, по которому сервер соединяется с Интернетом, синхронизирует время с помощью *ntpdate*, а затем разрывает соединение.

Следующий шаг – синхронизировать домашние компьютеры с сервером. Я не могу использовать для этого *ntpdate* (это ведь не демон), у меня работает *ntpd*. Я наконец разобрался, как его настроить – документация к NTP выглядит как диссертация по астрофизике, а не как руководство пользователя – и действительно могу синхронизировать компьютеры с сервером, но *ntpd* сам пытается синхронизироваться с Интернет-серверами каждые несколько минут, а так как соединения при этом нет, то он завершается аварийно.

Как можно настроить *ntpd*, чтобы он синхронизировался с Интернет-сервером только в заданное время? Или можно сделать так, чтобы для синхронизации времени использовалось мое задание с *ntpdate*, а сам *ntpd* раздавал на компьютеры локальное время сервера?

Мэтью Вайз



Если SUSE создаст ненужное соединение, *YaST* поможет его удалить.

О Дизайн большинства NTP-серверов, включая *ntpd* и *openntpd*, рассчитан на постоянное подключение к Интернету. Поддержка связи с другими серверами времени является неотъемлемой частью алгоритма их работы, что делает их бесполезными в Вашем случае.

Для обеспечения сервиса синхронизации с непостоянным или вообще отсутствующим Интернет-доступом создан *Chrony*. Он содержит две программы: *chronyd*, демон, раздающий клиентам локальное время сервера, и chronyc, который служит для синхронизации локального времени сервера с другими серверами времени. Последнюю версию пакета можно взять с **http://chrony.sunsite.dk**. Вам придется скомпилировать его из исходных текстов, но это простой процесс и он отлично описан в файле INSTALL.

Документация довольно многословная, однако простой сервер настроить очень легко. Просто поместите в **/etc/chrony.conf** следующие строки, заменяя **nnn.nnn.nnn.nnn** IP-адресом Вашего сервера:

```
server nnn.nnn.nnn.nnn offline
server nnn.nnn.nnn.nnn offline
server nnn.nnn.nnn.nnn offline
keyfile /etc/chrony.keys
commandkey 1
driftfile /etc/chrony.drift
allow 192.168.1
```

Список доступных серверов можно получить следующей командой:

```
netselect -s 3 pool.ntp.org
```

Параметр *offline* пресечет попытки Chrony синхронизироваться с Интернет-серверами. Команда *allow* задает диапазон IP-адресов, которым разрешено получать время с этого сервера. Установите пароль:

```
echo >/etc/chrony.keys `1
somepassword'
```

Теперь запустите демон с помощью init-скрипта: Ваш сервер будет раздавать компьютерам из домашней сети свое локальное время. Для корректного обновления локального времени лучше исправить *Cron*-задание, дописав в сценарий cron после соединения такие строки:

```
/usr/local/bin/chronyc <<EOF
password somepassword
online
EOF
```

Проделайте это еще раз после разрыва соединения, заменив online на **offline**. ***НБ***

Порулить мышью

В У меня есть IntelliMouse с семью кнопками: 1 = левая, 2 = средняя, 3 = правая, 4 и 5 = колесо, 6 и 7 = дополнительные кнопки. Я хотел бы контролировать этими кнопками уровень громкости, а может, переключение песен в *Amarok*. Насколько я понимаю, все это можно сделать, исправив файл **xorg.conf**. Можете помочь?

Тим

О Для начала нужно убедиться, что все семь кнопок посылают события в X-сервер. Для этого запустите **xev** и покликайте по его окну всеми кнопками. Если нажатие распознано, Вы увидите что-то вроде этого:

```
ButtonRelease event, serial 31,
synthetic NO window
0x3600001,
root 0x5a, subw 0x0,
time
191458267, (86,1 1),1
root:(91,162), state 0x1 button 4,
10, same_screen YES
```

Если сервер не получает событий от дополнительных кнопок, отредактируйте **xorg.conf**. У Вас уже должна быть строка **ZAxisMapping;** просто поменяйте в ней две цифры последних номеров кнопок и добавьте строку **Buttons**, указав в ней количество кнопок. Вот как это выглядит для семикнопочной мыши:

```
Section «InputDevice»
Identifier «USBMouse»
Driver «mouse»
Option «Protocol» «auto»
Option «Device» «/dev/input/mice»
Option «Buttons» «7»
Option «ZAxisMapping» «6 7»
EndSection
```

Перезапустите X-сервер и снова попробуйте **xev**. Ваши дополнительные кнопки – это 4 и 5, а колесо теперь будет 6 и 7. Сопоставим мышиным событиям нужные действия. Для этого можно взять *XBindKeys* (**http://hocwp.free.fr/xbindkeys**). У *XBindKeys* очень простой файл конфигурации (**~/.xbindkeysrc**). Например, он может выглядеть вот так:

```
«firefox»
 b:4
```

В данном случае при щелчке по кнопке 4 запустится *Firefox*. Для контроля любого KDE-приложения Вам нужно изучить DCOP (Desktop Communications Protocol). Запустите **kdcop** и посмотрите, какие команды принимает *Amarok*. Выполните эти команды из **kdcop**, и увидите, как их можно запустить из командной строки, скрипта или *XBindKeys*. Например, для перехода на следующую песню в *Amarok* нужно выполнить такую команду:

```
dcop amarok player next.
```

НБ

Почто-man

В На моем компьютере установлена Fedora Core 4, и я хотел бы знать, можно ли посылать письма из *Cron*-задания через запущенный *Sendmail*-сервер. Это спасло бы меня от запуска второй копии *Sendmail*.

Мэтт Уэлш

>>

О Несмотря на то, что для работы *Sendmail* или *Postfix* в Fedora Core 4 почти не требуется редактировать файлы конфигурации, Вы можете воспользоваться *ESMTP* или *SSMTP* для отправки почты через внешний почтовый сервер. Раньше я для этого пользовался *SSMTP*, однако сейчас, похоже, через Yum работает только *ESMTP*:

```
# yum install esmtp
# cat > /etc/esmtprc << «EOF»
hostname = mailserver:25
mda «/usr/bin/procmail -d %T»
```

Дополнительные кнопки мыши можно использовать для контроля KDE-программ через DCOP.

```
EOF
#
```

Эта простая конфигурация будет пропускать почту через сервер mailserver на 25-м порту. Рекомендую прочесть **man-страницы** для *esmtp* и **esmtprc** для ознакомления со всевозможными опциями.

Если *Sendmail* по умолчанию является MTA (запустите **alternatives --display mta**, чтобы это выяснить), это можно исправить, выбрав *ESMTP*:

```
# alternatives --config mta
```

Команда отобразит меню для выбора MTA по умолчанию.

Наконец, если Вы намерены пропускать через *Sendmail* даже почту для удаленных ящиков (скажем, направив выход *Cron* на адреса **@gmail.com**), убедитесь, что Вы верно отредактировали **/etc/mail/access**.

Как я уже говорил, вместо *ESMTP* можно использовать *SSMTP*. Он не доступен через Yum, так что Вам нужно будет установить его вручную:

```
# cd /root
# wget ftp://ftp.debian.org/debian/pool/main/s/ssmtp/ssmtp_2.61.orig.tar.gz
# tar -xzvf ssmtp_2.61.orig.tar.gz
# cd ssmtp_2.61
# make
# make install
```

Скрипт устроит Вам небольшой допрос и установит *SSMTP*-сервер в **/usr/local/sbin**, а файл его конфигурации (**ssmtp.conf**) – в **/usr/local/etc/ssmtp/**. Проверьте в файле конфигурации строку mail: она должна указывать на Ваш *Sendmail*-сервер. Для получения детальной информации запустите **man ssmtp** и рассмотрите пример конфигурационного файла в директории **ssmtp_2.61** (содержащей исходные коды).

Так как Вы устанавливали программу не через менеджер пакетов, а вручную, нужно добавить *SSMTP* в список доступных MTA:

```
# alternatives --install /usr/sbin/sendmail mta /usr/local/sbin/ssmtp 10
```

Теперь убедитесь, что *SSMTP* является MTA по умолчанию:

```
# alternatives --config mta
```

Не забудьте, что если Вы хотите пропускать почту для внешних адресов, то надо соответствующим образом настроить *Sendmail*. ***КК***

Бондинг

В У меня на компьютере под управлением Red Hat установлены две сетевые карты Ethernet, которые подключены к одному коммутатору (switch), и у меня один статический IP-адрес. Можно ли без виртуального IP настроить оба сетевых интерфейса как два параллельных пути, таким образом, чтобы в случае обрыва одной связи трафик переключался на другую?
Тони

О Да, бондинг допускает избыточность: один хост подключается к сети через два независимых пути (интерфейса). В отличие от метода с виртуальным IP-адресом, бондинг создает «плавающий» виртуальный интерфейс. Под Red Hat Вам нужно ассоциировать два физических интерфейса с одним виртуальным, 'bond0', при помощи стандартных файлов сетевой конфигурации. Итак, **ifcfg-eth0** и **ifcfg-eth1** должны содержать следующее:

```
/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
ONBOOT=yes
TYPE=Ethernet
MASTER=bond0
SLAVE=yes
/etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE=eth0
ONBOOT=yes
TYPE=Ethernet
MASTER=bond0
SLAVE=yes
```

Теперь создайте файл сетевого интерфейса **bond0**, который должен содержать что-то вроде этого:

```
/etc/sysconfig/network-scripts/ifcfg-bond0
DEVICE=bond0
ONBOOT=yes
BOOTPROTO=static
TYPE=Ethernet
IPADDR=x.y.z.a
NETMASK=x.y.z.a
```

Этим вы ассоциировали два физических интерфейса с одним виртуальным.

Вам нужно настроить запуск бондинга при загрузке. Для этого добавьте следующие строки в **/etc/modprobe.conf**:

```
alias bond0 bonding options
bonding mode=1 miimon=100
primary=eth0
install bond0 /sbin/modprobe eth0; /sbin/modprobe eth1; /sbin/modprobe bonding; /bin/true
```

mode=1 – активный/неактивный
miimon=100 – период опроса сети в миллисекундах (100мс)

Теперь загрузите модуль бондинга и перезапустите сетевой init-скрипт. Выполните следующие команды от лица суперпользователя:

```
# modprobe bond0
# service network restart
```

Последнее, что осталось сделать – обновить конфигурацию *iptables* и заменить упоминания о реальных интерфейсах на виртуальный. ***КК***

Адский RPM

В Я пробовал установить *OpenOffice.org* с DVD номера *LXF74*. Разархивирование z-файла с DVD не обнаружило никакого **./configure**, только RPM-файлы. Используя *rpm* в командной строке, я обнаружил порочный круг: **CORE01** требует для установки пакеты CORE02-CORE08, а пакеты **CORE02–CORE08 – CORE01**.

У меня дистрибутив Mandrake 9.1, установленный с CD номера *LXF41*. *Rpm* я запускал из домашней директории, в которую скопировал пакеты.
Джон

О Как Вы запускали *rpm*? Если Вы пытались устанавливать пакеты по очереди, то понятно, почему у Вас ничего не вышло – они зависят друг от друга. Команда *rpm* может разрешить эту ситуацию, но только если Вы укажете все пакеты сразу:

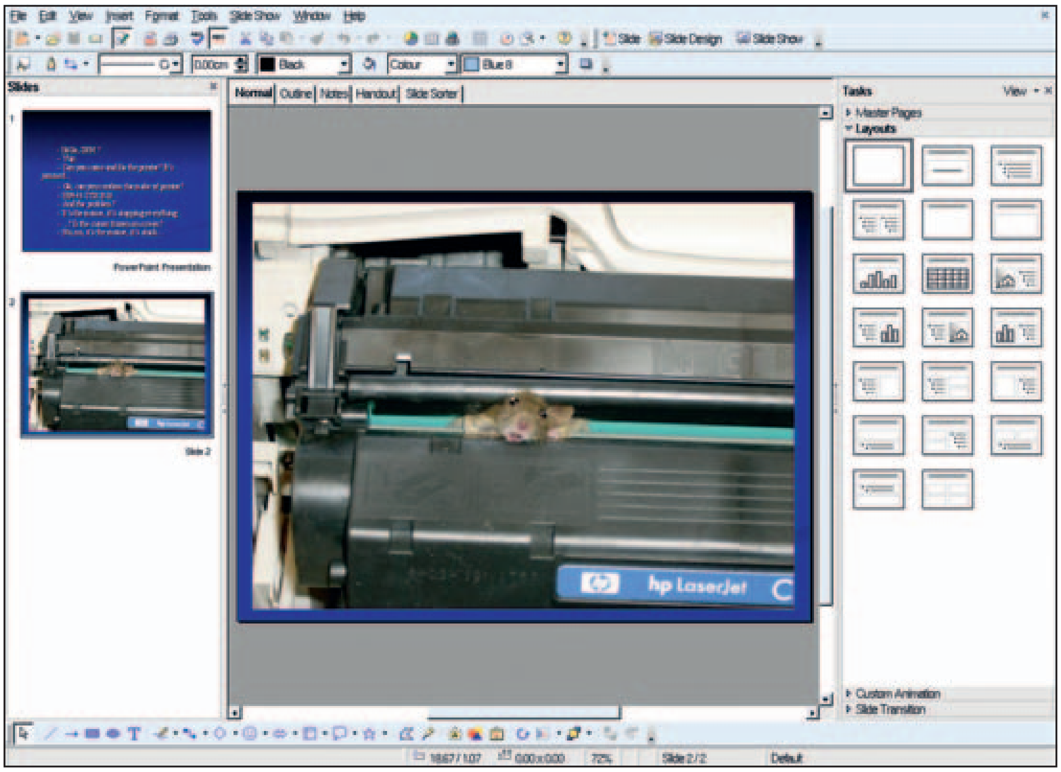
```
rpm -Uhv *.rpm
```

Нужно лишь позаботиться, чтобы в директории, где Вы запускаете эту команду, не было других *RPM*-пакетов. ***НБ***

Примочки QuickTime

В Я установил SUSE 10.0 с DVD номера *LXF74*: система очень проста в установке и содержит бесплатный офисный пакет, однако я не смог найти подключаемый модуль *QuickTime* для *Firefox*. Пробовал Quicktime4Linux – но с тех пор, как вышел *QuickTime 7*, он больше не работает. На сайте Apple подразумевается, что у меня либо Windows, либо MacOS.

В любом случае, трудно поверить, что Apple не портировала модуль на Linux, ведь OS X использует ядро Linux. Если и вы не знаете альтернативного сайта с трейлерами, я оказываюсь в луже.
Джефф Баркл



***OpenOffice 2.0* устанавливается из нескольких RPM-пакетов, их надо установить одновременно (в любом дистрибутиве).**

О Подключаемый модуль *MPlayer* для *Mozilla* работает также и с *Firefox* (**http://mplayerplugin.sf.net**). Он позволит Вам просматривать мультимедиа-файлы прямо в браузере. И в формате *QuickTime* тоже – правда, не все. Множество новых трейлеров сжаты закрытым кодеком Sorensen, который не поддерживается ни одним открытым проектом – пока не будет взломан.

Есть и другой вариант: с помощью *CrossOver Office* (разновидность Wine) можно использовать Windows-модули в Linux-браузерах, а также запускать приложения Windows прямо из-под Linux. *CrossOver Office* доступен на **www.codeweavers.com/site/products/cxoffice**. Стандартная версия стоит $39.95.

Между прочим, Mac OS X базируется не на ядре Linux, а на разновидности BSD [а если быть еще точнее – на микроядре Mach и некоторых пользовательских утилитах BSD, – прим. ред.]. ***НБ***

Promise неправ?

В Я использую CentOS (основанный на RHEL), и не могу заставить работать SATA-контроллер Promise, к которому присоединен 500-ГБ диск. Последняя версия драйвера работает только со старым ядром. Я писала письма в Promise 20–30 раз, и вместо помощи получила только стандартный электронный ответ.

А еще я планирую в скором времени купить ноутбук IBM, и хочу установить туда Linux. Можете ли вы сказать, какой дистрибутив поддерживает ноутбуки IBM лучше всего, со всякими там драйверами?

Жанна

О На моем компьютере установлен такой контроллер, и он нормально работает. На сайте выложены лишь старые версии драйвера по причине того, что новые версии теперь включены в ядро. Есть вероятность, что ядро было собрано без поддержки этого контроллера, а это значит, что Вам придется его пересобрать.

Если Вы никогда этого не делали, процесс может показаться сложным, но на самом деле все просто. Главное, не заменяйте старое ядро, сохраните его наряду с новым, тогда у Вас будет резерв на случай сбоя. Есть множество HOWTO насчет компиляции ядра, например, **www.digitalhermit.com/linux/Kernel-Build-HOWTO.html**.

Что касается дистрибутива для ноутбука IBM, то сгодится любой: оборудование IBM поддерживается Linux лучше всего, так что выбор зависит только от Вашего личного вкуса. Многие дистрибутивы имеют LiveCD-версии, которые можно запустить без установки на жесткий диски и посмотреть, какой Вам симпатичнее. Рекомендую ознакомиться с рубрикой Distrowatch. ***НБ***

В *Firefox* есть подключаемые модули для большинства медиа-форматов, но некоторые проприетарны. Пока единственное решение – *CrossOver Office*.

Редкостный RAID

В Имеется двухпроцессорный сервер Intel со SCSI-контроллером Adaptec 7899 (использует драйвер aic7xxx.o), на котором установлен RedHat AS 2.1 с ядром ...smp-2.4.9-e27.i386. Имеется внешний дисковый RAID-массив фирмы CNSi Chapparal G6322. По-умолчанию, при загрузке сервера, внешний RAID-массив не определяется – это видно из файла **/var/log/messages**.

Как сделать возможным использование данного RAID-массива на Linux-сервере?

Доподлинно известно, что «прикрутить» внешний RAID-массив возможно (второй такой же сервер работает с этим Chapparal G6322). После тщательного поиска на работающем сервере обнаружил в исходниках ядра (явно изменённого – сравнивал) файл **scsi_scan.c** в котором имеется строка с CNSi Chapparal G6322.

Я понимаю, что можно сделать BACKUP системы с данного сервера и восстановить на «подопытном». Но это не выход т.к. всё равно ядро нужно обновлять, чтобы работал подправленный *Oracle 9i*.

Николай Дмитриев

О В первую очередь, я бы порекомендовал вам не использовать модифицированные ядра при работе с базами данных *Oracle*, так как при этом вы теряете сертификацию для платформы и техническую поддержку от *Oracle* и *Red Hat*. Далее, обновите версию ядра до последней на сегодняшний момент – kernel-smp-2.4.9-e.68.i686.rpm. Затем у вас не должно будет возникнуть проблем с самим SCSI-контроллером Adaptec 7899, который должен работать в RHEL 2.1 «из коробки». Скорее всего, вы столкнулись с известной особенностью RAID-массивов Chapparal связанной с нестандартной нумерацией LUN-ов. В RHEL версии 4 и выше вы могли бы попробовать воспользоваться утилитой *scsi_id*, но поскольку UDEV, через который работает данная утилита, не используется в RHEL 2.1, попробуйте воспользоваться ссылкой **http://www.garloff.de/kurt/linux/scsidev/**. При помощи скрипта **rescan-scsi-bus.sh** вы можете принудительно опросить шину на предмет подключенных устройств. Вызов скрипта нужно разместить в **/etc/rc.d/rc.sysinit** до момента проверки файловых систем утилитой *fsck*. ***AM***

KDE без проблем

В У меня возникла такая проблема: уже четвертый раз не получается установить KDE 3.5. Пробовал и KDE 3.5.1, обе системы – с дисков журнала.

Каждый раз выдает в самом конце:

★ВОПРОС-ПОБЕДИТЕЛЬ!

Редакция вместе с компанией Bitrix выбрали самый интересный вопрос месяца и поздравляют победителя с заслуженной наградой – сертификатом на 1000 рублей в LinuxCenter.ru

Да где же оно?

В Не могли бы вы подсказать мне хорошую программу-каталогизатор для CD под Linux? Что-нибудь наподобие программы *WhereIsIt?*, которую я когда-то видел под Windows. Основные требования: программа должна уметь сканировать компакт-диск и запоминать структуру каталогов, список всех файлов, с указанием их атрибутов – даты и времени создания и осуществлять удобный поиск по имени файла в своей базе. Также большим плюсом было бы запоминание не только имен файлов, но и содержимого архивов.

Я давно ищу подобную программу, и к сожалению ничего подобного так и не встречал.

Сергей

О Попробуйте программу *Gwhere* (**http://www.gwhere.org/**), которая упоминается в разделе HotPicks февральского номера Linux Format. Она удовлетворяет всем перечисленным вами требованиям и даже умеет осуществлять поиск внутри архивов, правда, для этого может потребоваться пересобрать ее из исходных текстов. Кстати, программа доступна не только для Linux, но и для Windows.

СК

Комментарий редакции: нам было особенно приятно опубликовать эту пару «вопрос-ответ». Дело в том, что Сергей не только сам нашел решение поставленной им задачи быстрее, чем это сделали мы, но и не поленился описать его и отправить в журнал. За стремление к читательской взаимовыручке (в духе Open Source) и самостоятельность в решении задач мы с удовольствием награждаем Сергея почетным титулом автора «Вопроса месяца»!

LINUX FORMAT

bitrix

```
grep: /usr/X11R6/lib/libGL.la: No such file or directory
/bin/sed: невозможно прочитать /usr/X11R6/lib/libGL.la: No such file or directory
libtool: link: `/usr/X11R6/lib/libGL.la' is not a valid libtool archive
make[5]: *** [kcm_info.la] Ошибка 1
make[5]: Leaving directory `/opt/kde3.5/konstruct/kde/kdebase/work/kdebase-3.5.0/kcontrol/info'
make[4]: *** [all-recursive] Ошибка 1
make[4]: Leaving directory `/opt/kde3.5/konstruct/kde/kdebase/work/kdebase-3.5.0/kcontrol'
make[3]: *** [all-recursive] Ошибка 1
make[3]: Leaving directory `/opt/kde3.5/konstruct/kde/kdebase/work/kdebase-3.5.0'
make[2]: *** [all] Ошибка 2
make[2]: Leaving directory `/opt/kde3.5/konstruct/kde/kdebase/work/kdebase-3.5.0'
make[1]: *** [build-work/kdebase-3.5.0/Makefile] Ошибка 2
make[1]: Leaving directory `/opt/kde3.5/konstruct/kde/kdebase'
make: *** [dep-../../kde/kdebase] Ошибка 2
```

Из-за чего это? Подскажите, пожалуйста. Моя система: Mandriva 2006, AMD64 3000+; Abit KV8 Pro S754, 1,2Gb DDR400 RAM.

Александр Сергеевич

О Если для Вас не проблема выкачать из Интернета 200-300 Мбайт, то логичнее было бы обновить KDE из cooker-репозитория Mandriva Linux. В cooker довольно быстро появляются собранные пакеты самых свежих программ. Для того, чтобы обновить KDE из cooker, нужно добавить его в список репозиториев. Откройте Центр Управления Mandriva -> Управление программами -> Настройка источников. В менеджере источников нажмите кнопку «Добавить пользовательский» и введите в поле для URL **ftp://ftp.free.fr/pub/Distributions_Linux/MandrivaLinux/devel/cooker/amd64/media/main**. Подтвердите добавление источника и закройте центр управления. После этого обновить KDE можно командой **urpmi kdebase**.

Если же вы не имеете возможности скачать готовый пакет и намерены собрать KDE из исходных текстов, Вам следует установить пакет libMesaGLU1-devel, из-за отсутствия которого и происходит описанная Вами ошибка. ***АК***

Болтливый cron

В С cron'ом я вроде бы разобрался, но вот беда: как сделать так, чтобы я не видел его работы? А то каждый раз мне приходит письмо:

From: root@crd.example.net
Subject: cron: /home/user/script

А сценарий запускается каждые четыре минуты... Как быть?

С форума LXF Russia

О Чтобы cron не рассылал вам ежеминутный отчет о проделанной работе, отредактируйте свой файл crontab и укажите в качестве значения переменной окружения MAILTO пустую строку (MAILTO=»»). Имейте в виду, что отсутствие переменной MAILTO и пустая переменная MAILTO – это разные вещи. Кроме того, рекомендую Вам перенаправить стандартные потоки вывода и ошибок в **/dev/null**:

```
4 * * * * /home/user/script >/dev/null 2>&1
```

ВС

СПРАШИВАЙТЕ ПРАВИЛЬНО!

- Пожалуйста, не забывайте сообщать все необходимые данные о вашей системе. «У меня не работает X» нам мало что скажет, если мы не знаем, какую версию X вы имеете в виду и на какой аппаратной конфигурации пытаетесь ее запустить.
- Опишите свою проблему с максимальной точностью. Причитания типа «Оно не работает» или «У меня ошибка» вряд ли дадут нам возможность помочь Вам. Каким именно образом нечто не работает? Чего вы от него хотели? Какое поступает сообщение об ошибке?
- Помните, пожалуйста, что люди, работающие в нашем журнале, НЕ являются авторами или разработчиками Linux или Вашего конкретного пакета (дистрибутива). Иногда нужная Вам информация имеется на соответствующем сайте. Ознакомьтесь с документацией!

Мы стараемся ответить на все вопросы. Если вы не нашли ответа на свой, просмотрите другие ответы – может быть, там разбирается проблема, аналогичная вашей. Все вопросы, к сожалению, поместить невозможно.

Информация о диске

Внимательно прочтите это перед тем, как использовать DVD-диск.

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;
имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;
имя_программы-1.0.1.tar.gz – обычно это исходный код;
имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;
имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;
имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru

Содержание LINUX FORMAT DVD Апрель 2006

Сторона 1

Сторона 2

ASP Linux 11

IBM WebSphere Community Edition

СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование cdrecord, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку «Burn» и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на «Combust!». Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecrod* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

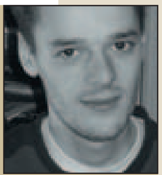
А что если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не придется использовать Linux для записи дисков, подойдет любая операционная система, способная распознать привод записи дисков (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика. LXF

Диск Linux Format

Заставьте SUSE преодолеть звуковой барьер, используя мощь дистрибутива Slick...

Linux или GNU/Linux?



Майк Сондерс чувствует себя неудачником, если устанавливает меньше двух дистрибутивов за день.

Как назвать вашу ОС: Linux или GNU/Linux? Спор будет длиться вечно, независимо от степени изнуренности оппонентов, а за этот месяц я читал о нем больше, чем когда-либо. С одной стороны, вполне понятно, что Ричард Столлмен (Richard Stallman) и Фонд Свободного ПО (Free Software Foundation) хотят признания и своих заслуг, и философии GNU, известной не менее широко, чем прагматизм Торвальдса. Иногда мы забываем, насколько важна свобода, в погоне за дешёвым и надёжным ПО.

С другой стороны, сколько современных дистрибутивов поддерживают философию GNU? Одно из исследований задалось целью подсчитать долю строк кода из проекта GNU в отдельно взятом дистрибутиве. Результат показал: код GNU занимает порядка 15%. Однако значимость GNU здесь ни при чем: в монстрах вроде *OpenOffice.org* десятки миллионов строк кода, но это не повод называть операционную систему 'OOo/Linux'.

Некоторые указывают на GCC и *Glibc* как на примеры вклада GNU. Но позвольте, эти разработки обязаны популярностью Linux. Если бы Линус не принялся за ядро системы, мы бы всё ещё ждали появления The Hurd, а GCC и *Glibc* были бы бледной тенью теперешних. GNU послужило основой, но именно Linux способствовал разработке многочисленных проектов GNU – и давайте не будем махать руками на тех, кто для краткости пропускает 'GNU' в имени.

mike.saunders@futurenet.co.uk

СИСТЕМА

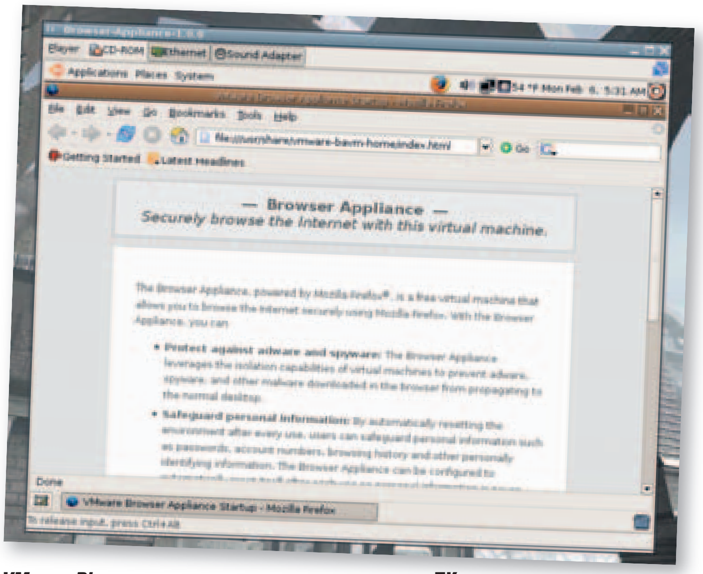
VMWARE PLAYER

Хотите запускать другие операционные системы на своём компьютере без мороки с установкой и переразбивкой жёсткого диска? *VMware Player* – решение данного вопроса, в чем вам легко убедиться самим, поскольку в этом месяце мы имеем эксклюзивную возможность добавить его Linux-версию на наш диск. *VMware Player* полностью эмулирует работу ПК, предоставляя виртуальный процессор, жёсткий диск, видеокарту и т.д., так что операционная система, запущенная поверх него, не может отличить виртуальный компьютер от реального. Это идеальное решение для тестирования дистрибутивов Linux – да и других ОС для ПК – прямо с любимого рабочего стола Linux.

В *VMware Player* вы просто открываете виртуальный образ диска (то есть файл), и окно, отобразившееся на вашем рабочем столе, будет действовать как виртуальный монитор. Теперь можете запустить операционную систему с виртуального образа диска, как на настоящем ПК. В отличие от традиционных эмуляторов типа *Bochs*, *VMware* использует технологии виртуализации современных процессоров семейства x86 для непосредственного выполнения команд процессора. Эмуляция команд работала бы очень медленно, а благодаря виртуализации скорость у *VMware* поразительная.

Чтобы установить его, распакуйте файл **VMware-player-1.0.1-19317.tar.gz** из раздела System на нашем диске, и запустите от имени суперпользователем скрипт **vmware-install.pl** в окне терминала. Он задаст вам несколько вопросов (ответов по умолчанию будет достаточно) и попытается установить специальный модуль ядра, ответственный за возможности виртуализации. Если для вашей системы модуля не обнаружится, установите заголовочные файлы ядра и перезапустите установку.

Когда она закончится, новая операционная система будет к вашим услугам. Чтобы ею воспользоваться, запустите **vmplayer** и откройте образ виртуального диска, например, Syllable, из файла **syllable-0.6.0a.zip** в разделе Distros на нашем диске. Попробуйте также скопировать каталог **Distros/DeLiLinux** на ваш жёсткий диск и загрузить файл **DeLi.vmx** – увидите усеченный настольный дистрибутив.



VMware Player **предоставляет полную эмуляцию ПК; здесь показан *The Browser Appliance*.**

ЗОНА ОСОБОГО ВНИМАНИЯ

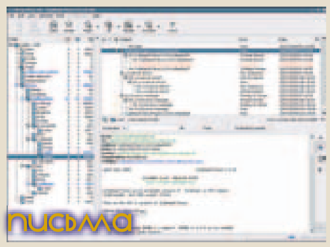
ТАКЖЕ НА ДИСКЕ

Не забудьте, что диск Linux Format в этот раз – двусторонний. Переверните «пластинку», чтобы добраться, может быть, до самого нужного именно вам.

Итак, в этом месяце на нашем диске кроме всего прочего присутствуют:

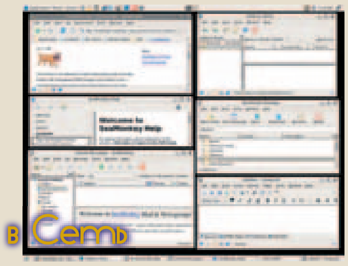
Sylpheed-Claws

Почтовая программа



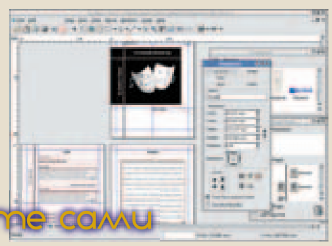
SeaMonkey

интернет-комбайн



Scribus

Издательская система



ИНТЕРНЕТ

THE BROWSER APPLIANCE

Вместе с *VMware Player* в этом выпуске мы добавили и *The Browser Appliance* - отличный способ улучшить безопасность Windows-машин с помощью Linux. Да, именно так: улучшить безопасность Windows с помощью Linux! Работает просто изумительно: устанавливаете Windows-версию *VMware Player* и открываете в нём образ жёсткого диска *Browser Appliance*, содержащего усеченный вариант Ubuntu. В результате у вас будет виртуальный ПК на Windows-машине, с работающим Linux, и не надо переразбивать диск.

Это здорово само по себе, но образ диска *Browser Appliance* разработан специально для того, чтобы запускать браузер *Firefox*. Поскольку web-браузер – наиболее частая цель атак на машинах Windows, имеет смысл использовать Linux, выходя в Интернет. Так что если на ПК вашего друга или соседа – рассадник вирусов и троянов, дайте ему *The Browser Appliance*: количество звонков в службу техподдержки сразу уменьшится. Заодно внесете свою лепту в пропаганду Linux.

Чтобы запустить *VMware Player*, установите его под Windows (хоть это и ПО для Windows, на нашем диске он есть) и откройте через него файл **Browser-Appliance.vmx** из архива **Browser-Appliance-1.0.0.zip**. После загрузки сможете использовать Gnome и *Firefox* как на обычном ПК. А главное – Windows будет в безопасности!

ДИСТРИБУТИВЫ

OPENSUSE SLICK

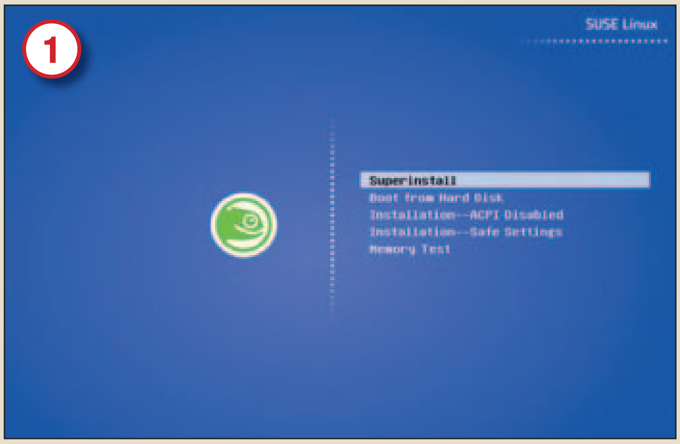
Дистрибутив этого месяца – ответвление от проекта OpenSUSE. Как только Novell приступила к открытию разработки SUSE, многие из постоянных фанатов этого дистрибутива начали работать над его настройкой и обновлением.

Вполне естественно, что если нововведения разработчиков слишком резко отошли от текущего поддерживаемого релиза SUSE, то лучше создать новую ветку. Так и появились Super и Slick, два ответвления SUSE, задавшиеся целью улучшить быстродействие и возможности дистрибутива, оставаясь по-прежнему отполированными – качество, которым славится SUSE .

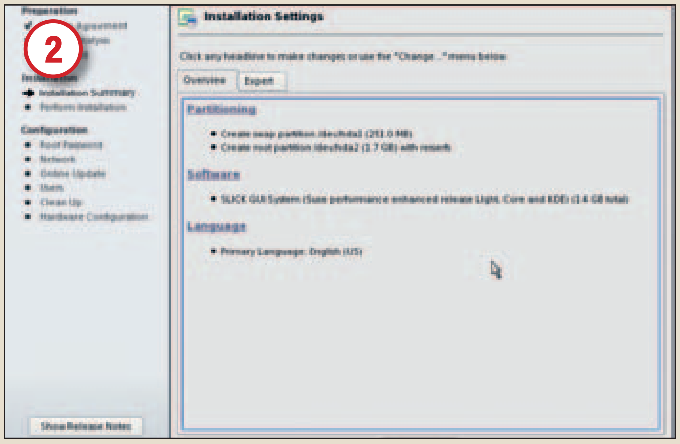
Объединенные усилия обоих вариантов породили однодисковые версии SUSE, где придирчиво отобранные программы соседствуют с обычными SUSE-утилитами, вроде Yast. Slick предоставляет KDE 3.4.3, *X.org 6.8.2* и *Firefox 1.5b*; улучшения производительности включают ускоренную загрузку, пре-компоновку (чтобы разделяемые библиотеки загружались быстрее) и ядро с заплатками, оптимизирующими функционирование рабочего стола. Кроме того, с помощью *Klick* (см. Hotpicks в *LXF74/75* или **http://klik.atekon.de**) вы сможете устанавливать программы с невероятной легкостью. А на нашем DVD – около 1000 дополнительных пакетов SUSE в каталоге **Distros/SLICK/Packages**.

Установка OpenSUSE Slick не сложнее установки любого другого варианта SUSE. Вот вам мини-руководство:

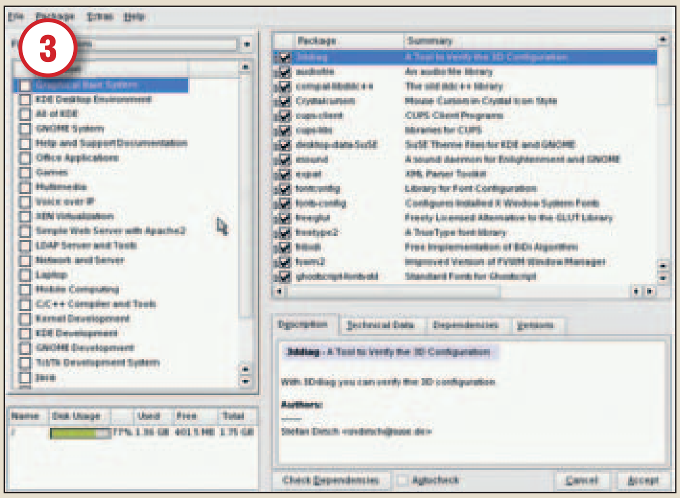
УСТАНОВКА OPENSUSE SLICK



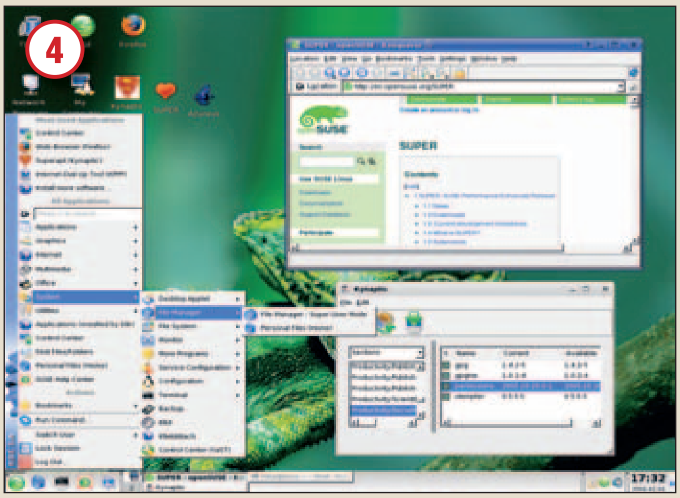
Вставьте DVD диск в DVD привод и перезагрузите ПК. (Понадобится настроить загрузку с DVD в вашем BIOS) На показанном экране, выберите Superinstall для запуска процедуры установки. Загрузка инсталлятора может занять несколько минут.



Инсталлятор определит наилучшие настройки для вашей машины автоматически. Кликните на Partitioning, если хотите по-своему разбить жёсткий диск. Рекомендуется отвести разделу для Linux не менее чем 2 ГБ. Будьте осторожны, чтобы по ошибке не уничтожить имеющиеся разделы Windows.



Вы можете сами выбирать, какие программы установить, хотя настройки по умолчанию идеальны для настольной машины. Выполняя ваши изменения, следите за индикатором Disk Usage в левом нижнем углу экрана, чтобы не выйти за пределы отведенного дискового пространства. Нажмите Accept, и начнется процесс копирования файлов на ваш жёсткий диск.



После нескольких финальных вопросов о конфигурации (запишите выбранные пароли!) система перезагрузится: уберите диск из DVD-привода, и ваш ПК загрузится в привычный рабочий стол KDE. Чтобы изменить настройки, запустите *Yast* (из пункта System главного меню). Наслаждайтесь!

>>

<< ГРАФИКА

ПО ДЛЯ ЦИФРОВОЙ КАМЕРЫ

В соответствии с темой номера, мы подготовили набор лучших средств для работы с цифровым фото в Linux. Скачиваете ли вы фотографии с цифровой камеры, редактируете ли их в графическом редакторе или подготавливаете слайды – на нашем диске в разделе Graphics найдутся хорошие программы-помощницы.

Как всегда, на диске записан исходный код, так что собрать их вы сможете сами. По возможности включены также бинарные пакеты (типа .deb). Если программа потребует ПО, которое обычно не включено в большинство дистрибутивов, вы найдёте его в каталоге dependencies самого приложения.

РАБОЧИЙ СТОЛ

GNOME 2.14

К тому моменту, когда вы будете читать эти строки, бинарные пакеты Gnome наверняка появятся в репозитарии используемого вами дистрибутива. Мы же ограничимся наиболее общим вариантом – исходными текстами. Чтобы облегчить компиляцию и сборку, используйте систему **GARNOME**, которая также есть на нашем DVD. Распакуйте файл **garnome-2.14.0.tar.bz2** в какой-нибудь каталог на вашем жестком диске, отредактируйте параметры в файле **gar.conf.mk**. В первую очередь, исправьте переменную **GARCHIVEDIR**, чтобы она указывала на директорию, в которой хранятся исходные тексты GNOME. В нее же, при необходимости, следует поместить библиотеки из каталога Essentials: *Gtk+*, *Glib*, *LibXML* и другие. Затем обратите внимание на переменную **main_prefix** – в ней хранится название каталога, в который будет установлен свежесобранный Gnome. По умолчанию, это - **$HOME/garnome**, но вы можете изменить путь на **/opt/gnome-2.14** или подобный, чтобы предоставить доступ к рабочему столу для всех пользователей системы. Теперь можно приступать к сборке.

Дополнительную информацию можно почерпнуть из readme-файла.

Похорошевший GNOME 2.14 в Ubuntu Drapper.

ДРУГИЕ ПРОГРАММЫ

В этом месяце прямо-таки обрушилась лавина новых релизов, от программ для рабочего стола и сервера до инструментов разработки и работы со звуком.

Возможно, наиболее значительное событие – первый релиз *SeaMonkey*, последователя *Mozilla Suite*. Когда в Mozilla Foundation решили сфокусировать усилия на *Firefox* и *Thunderbird*, исходный пакет *Mozilla Suite* оказался на задворках. Какое-то время он обновлялся, чтобы поддержать использующих его людей, но будущего у него не было.

К счастью, сообщество вступилось за этот проект и дало ему вторую жизнь под именем *SeaMonkey* (первое имя браузера Mozilla). *SeaMonkey* включает браузер, почтовый клиент, редактор страниц и IRC-клиент – всё в одном приложении. Вы можете спросить: а кому это надо, раз уж так популярен *Firefox*? Ну, если *Firefox* и *Thunderbird* запущены у вас как отдельные приложения, значит, и версий движка *Gecko* тоже две, а это перерасход памяти.

Не считая новой иконки, SeaMonkey – вылитый Mozilla Suite.

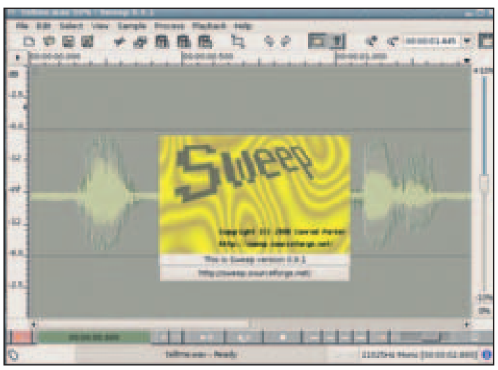
SeaMonkey менее требователен к ресурсам, если вы используете все его приложения, и к тому же радует своей стабильностью.

Если ваш почтовый клиент неуклюж и вы ищете что-нибудь полегче, попробуйте *Sylpheed-Claws* из раздела Internet. Этот клиент ворвался в мир, как супер-новая версия *Sylpheed*, и с тех пор приобрёл верных поклонников, считающих его улучшенной инкарнацией родительского приложения. Отличный выбор, если вы чувствуете себя неуютно в *Evolution* или *KMail*.

После двух лет работы и более 250 улучшений, вышел новый релиз известной утилиты сканирования *Nmap* (известна тем, что снималась в фильме *Матрица: перезагрузка*). Новые возможности *Nmap 4.0* включают ARP-сканирование, улучшенное распознавание протоколов, новую систему распознавания ОС, новый интерфейс на базе *GTK 2* и множество улучшений по скорости.

Man-страница *Nmap* переписана заново, да есть еще и волнующий сайт **www.insecure.org**.

Sweep, мощная программа редактирования и воспроизведения звука, достигла версии 0.9.0. Она имеет впечатляющий набор возможностей, включая многопоточный фоновый обсчёт, поддержку расширений LADSPA и DJ-функциональность – достойный выбор, если вы работаете со звуком в Linux.



Sweep 0.9.0 – первый релиз, поддерживающий GTK 2, и его графический интерфейс наконец-то похорошел.

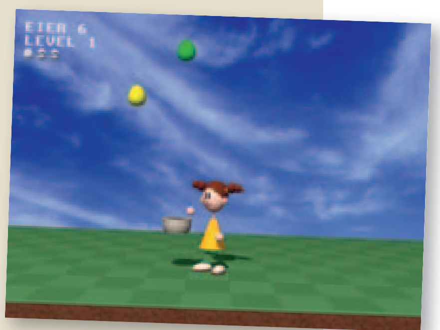
И НАКОНЕЦ...

ИГРЫ

Хотя ролевые игры, возможно, такая же древность, как холмы, по которым вы часто в них гуляете, этот жанр никогда не умрёт. *GalaxyMage*, едва достигшая версии 0.2.1, но уже неплохо заявившая о себе – пошаговая трёхмерная стратегия для Linux, Windows и Mac OS X. В отличие от других игр, *GalaxyMage* сделана по принципу «бери-и-играй»: не нужно заучивать команды и комбинации клавиш, не тратится время на нудный сюжет. Чтобы поиграть, просто разархивируйте файл **galaxymage-0.2.1.tar.gz**, и запустите **GalaxyMage.py** из вновь созданной директории (потребуются библиотеки *PyGame* и *PyOpenGL*).

Если вы любитель ретро, попробуйте *The Bub's Brothers 1.5*. По названию можно догадаться, что это современный вариант бессмертной игры Bubble Bobble, написанный на *Python* с помощью *PyGame*. Разархивируйте файл **bubbros-1.5.tar. bz2** и запустите **./BubBob.py** из созданного каталога. Не бойтесь, если этот скрипт запустит браузер – это для настройки многопользовательского режима игрового сервера. Вы можете запустить собственный сервер и для игры подсоединиться к нему с того же компьютера.

И наконец, мы просто не смогли сдать этот диск в тираж, не включив в него игру *Sarah-Maries Eierjagd*. Эта чарующая нелепостью игра (в переводе название означает: Сара-Мари ловит яйца) сопровождается текстом только по-немецки, но цель ее незамысловата: гоняйте девушку, чтоб она ловила яйца и копила очки – а если сдуру ухватите три тухлых яйца, игре конец. Игру написал Патрик Гердсмайер (Patrick Gerdsmeier) в подарок на Пасху своей племяннице по имени Сара-Мари. Сумеете ли вы побить рекорд автора – 392 очка? А сможете утерпеть и не попробовать эту игру самой первой? **LXF**



Все яйца в одну корзину – нерационально везде, кроме *Sarah-Maries Eierjagd*.

СОДЕРЖАНИЕ ДИСКА

LINUX FORMAT DVD

СТОРОНА 1

Рабочий стол

Gnome-RDP	Клиент удалённого доступа к рабочему столу.
GnomeBaker	Утилита записи CD/DVD.
GShowTV	Запись и просмотр TV-программ.
Gnome	Открытый рабочий стол.
KMyMoney	Персональный менеджер финансов.
Komparator	Синхронизация каталогов.
Scribes	Текстовый редактор Gnome.
SpeedCrunch	Высокоточный калькулятор.

Разработка

CControl	Контроллер компилятора.
Free Pascal	Язык программирования Pascal.
Lazarus	Бесплатная интегрированная среда разработки для Pascal.
OpenGUI	Библиотека для создания GUI.
Qt	Среда разработки GUI-приложений.
RText	Текстовый редактор для программистов.
Rubygame	Библиотека для разработки игр в языке Ruby.

Дистрибутивы

Deli Linux	Облегчённый дистрибутив (образ *VMware*).
OpenSUSE SLICK	Вариант SUSE с улучшенной производительностью.
Syllable	Уникальная настольная ОС.

Игры

Egg Hunt	Ловим яйца.
Enemy Lines 3	Абстрактная трёхмерная стрелялка.
GalaxyMage	Тактический/стратегический RPG.
GLPuzzle	Головоломки.
SDLHana	Игра Hanafuda.
The Bub's Brothers	Клон Bubble Bobble.
Wormux	Клон Worms

Графика

Утилиты для цифровой камеры:

DCRaw	Декодирование необработанных фотографий.
DigiKam	Утилита управления фотографиями.
F-Spot	Утилита управления фотографиями.
Gimp	Редактор изображений.
GPhoto2	Интерфейс командной строки к библиотеке Libgphoto2.
Gtkam	GTK-интерфейс к Libgphoto2.
Inkscape	Векторный графический редактор.
Kalbum	Менеджер фотоальбомов.
KFlickr	Утилита заливки во Flickr для KDE.
Libgphoto2	Библиотека доступа к фотокамере.
LittleCMS	Библиотека для управления цветностью.
QDVDAuthor	Графический интерфейс для DVDAuthor.
Scribus	Настольная издательская система
UFRaw	Манипулирование изображениями.

Интернет

Evolution	Персональный органайзер/e-mail клиент.
Firefox	Web-браузер.
Jingle	Библиотека для мгновенного обмена сообщениями.
KBlogger	Ведение блога с помощью апплета Kicker.
Nmap	Сетевой сканер.
SeaMonkey	Браузер и почтовый клиент.
Sylpheed-Claws	Почтовый клиент.

Сервер

Aterr	Форум с поддержкой тем.
DenyHosts	Блокируем атаки на SSH-сервер.
MySQL	Сервер баз данных.
Nagios	Мониторинг сети и серверов.
PhpFreeChat	Чат-сервер, использующий технологию Ajax.
Savane	Система хостинга.
Wiclear	Wiki-сервер.

Звук

AudConvert	Конвертер форматов аудио.
Muine	Проигрыватель музыки для Gnome.
Playground	Апплет управления музыкальным плейером.
Sweep	Редактор звуковых файлов.
TuxGuitar	Редактор гитарных таблиц.

Система

Fcron	Планировщик команд.
Fish	Командная оболочка.
KPowersave	Монитор аккумуляторов для KDE.
LinuxBIOS	BIOS с открытым кодом.
Shush	Оболочка для заданий Cron.
SlackFire	Создание брандмауэра для Slackware.
VMware Player	Запуск виртуальных ПК.

СТОРОНА 2

ASP Linux 11

IBM WebSphere Community Edition

***Wormux*: *Worms* для Linux!**

LINUX FORMAT

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно
Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:

ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru

Переводчики
Юрий Афанасьев, Александр Бикмеев, Светлана Кривошеина, Александр Кузьменков, Алексей Опарин, Сергей Салимов, Сергей Супрунов, Александр Толстой, Александр Черных, Юлия Шабунио, Павел Шер.
Литературные редакторы
Родион Водейко, Иван Мищенко, Елена Толстякова
Допечатная подготовка
Мария Пучкова
Родион Водейко
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Заместитель генерального директора
Софья Виниченко
Генеральный директор
Павел Фролов
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Павел Фролов, Станислав Медведев
Отпечатано в типографии "Текст",
Пре–пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Ник Вейтч (Nick Veitch) nick.veitch@futurenet.co.uk
Заместитель редактора Пол Хадсон (Paul Hudson) paul.hudson@futurenet.co.uk
Художественный редактор
Джулиан Джефферсон (Julian Jefferson) julian.jefferson@futurenet.co.uk
Новостной редактор Майк Сондерс (Mike Saunders) mike.saunders@futurenet.co.uk
Операционный редактор
Ребекка Смелли (Rebecca Smalley) rebecca.smalley@futurenet.co.uk
Штатный автор
Грэм Моррисон (Graham Morrison) graham.morrison@futurenet.co.uk
Ассистент по выпуску
Эндрью Грегори (Andrew Gregory) andrew.gregory@futurenet.co.uk

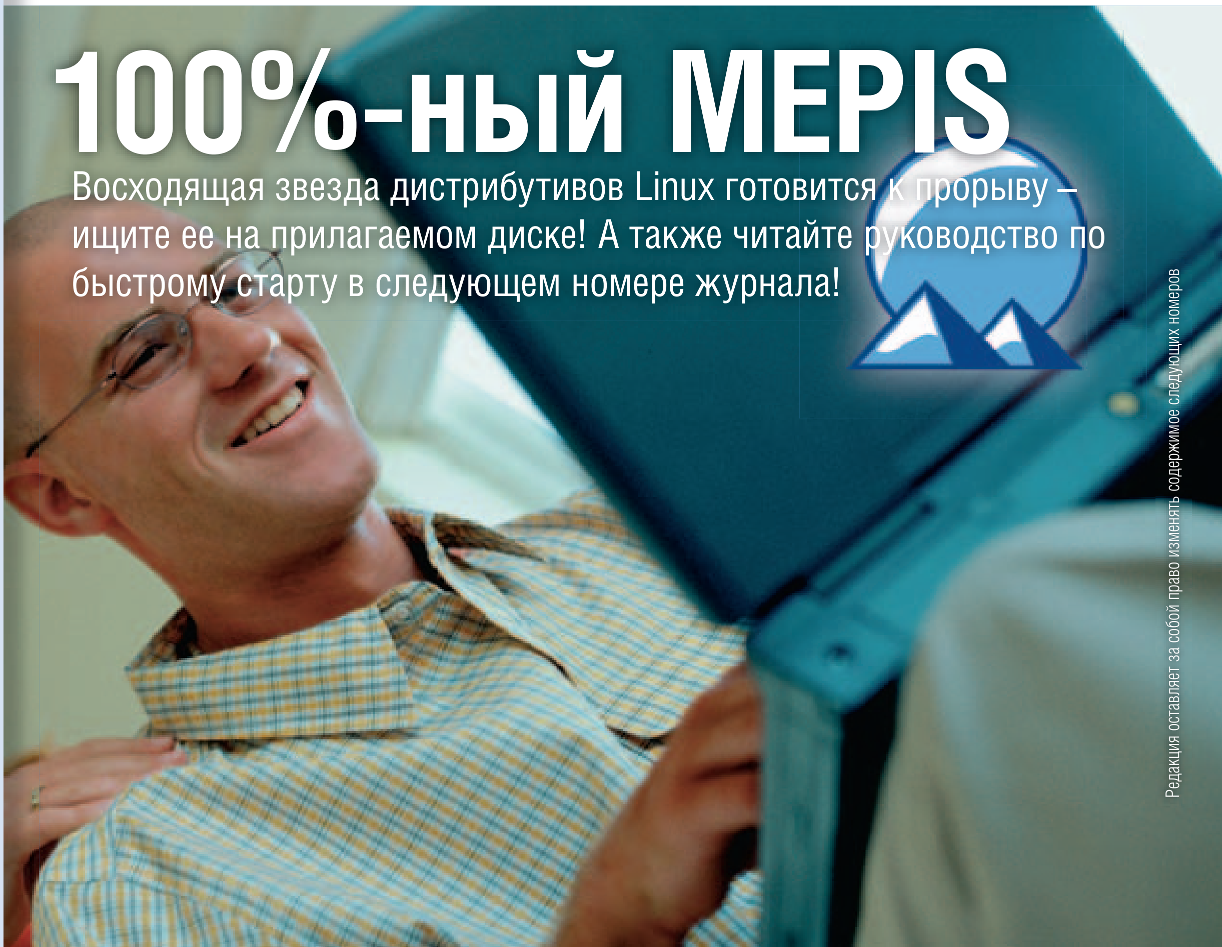
АВТОРЫ
Ладислав Боднар (Ladislav Bodnar), Нейл Ботвик (Neil Bothwick), Энди Чэннел (Andy Channelle), Дэвид Картрайт (David Cartwright), Ричард Коббетт (Richard Cobbett), Алекс Кокс (Alex Cox), Марко Фиоретти (Marco Fioretti), Джо Харрис (Jo Harris), Майкл Микс (Michael Meeks), Дмитрий Кирсанов, Александр Бикмеев, Александр Кузьменков, Владимир Попов, Сергей Супрунов, Сергей Яремчук

ХУДОЖЕСТВЕННЫЙ ОТДЕЛ
Арт-ассистенты: Джон Блекшау (John Blackshaw), Симон Годдард (Simon Goddard), Дэниэль Винсент (Daniel Vincent)
Фотографии: Joby Sessions, Simon Lees, Superstock, Photodisc, Photolibrary.com
Иллюстрации: Пол Бейтмен (Paul Bateman), Крис Винн (Chris Winn), Шейн Коллин (Shane Collinge)
Создание диска: Майк Сондерс (Mike Saunders)

КОНТАКТНАЯ ИНФОРМАЦИЯ
England: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email linuxformat@futurenet.co.uk
Россия: Санкт-Петербург, ул. Гончарная, 23, офис 54, телефон: (812) 717-00-37
Email: info@linuxformat.ru
Web: www.linuxformat.ru

В следующем месяце

LINUX FORMAT 5(79) МАЙ 2006



100%-ный MEPIS

Восходящая звезда дистрибутивов Linux готовится к прорыву – ищите ее на прилагаемом диске! А также читайте руководство по быстрому старту в следующем номере журнала!

Редакция оставляет за собой право изменять содержимое следующих номеров

LXF ИНТЕРВЬЮ

Брюс Пиренс

Гуру Debian и лидер движения свободного ПО

Создаем Linux-киоск

Сенсорный экран + Linux + это руководство = прибыль

Open Solaris

Альтернативная ОС от Sun Microsystems